

RasPi

DESIGN
BUILD
CODE

12

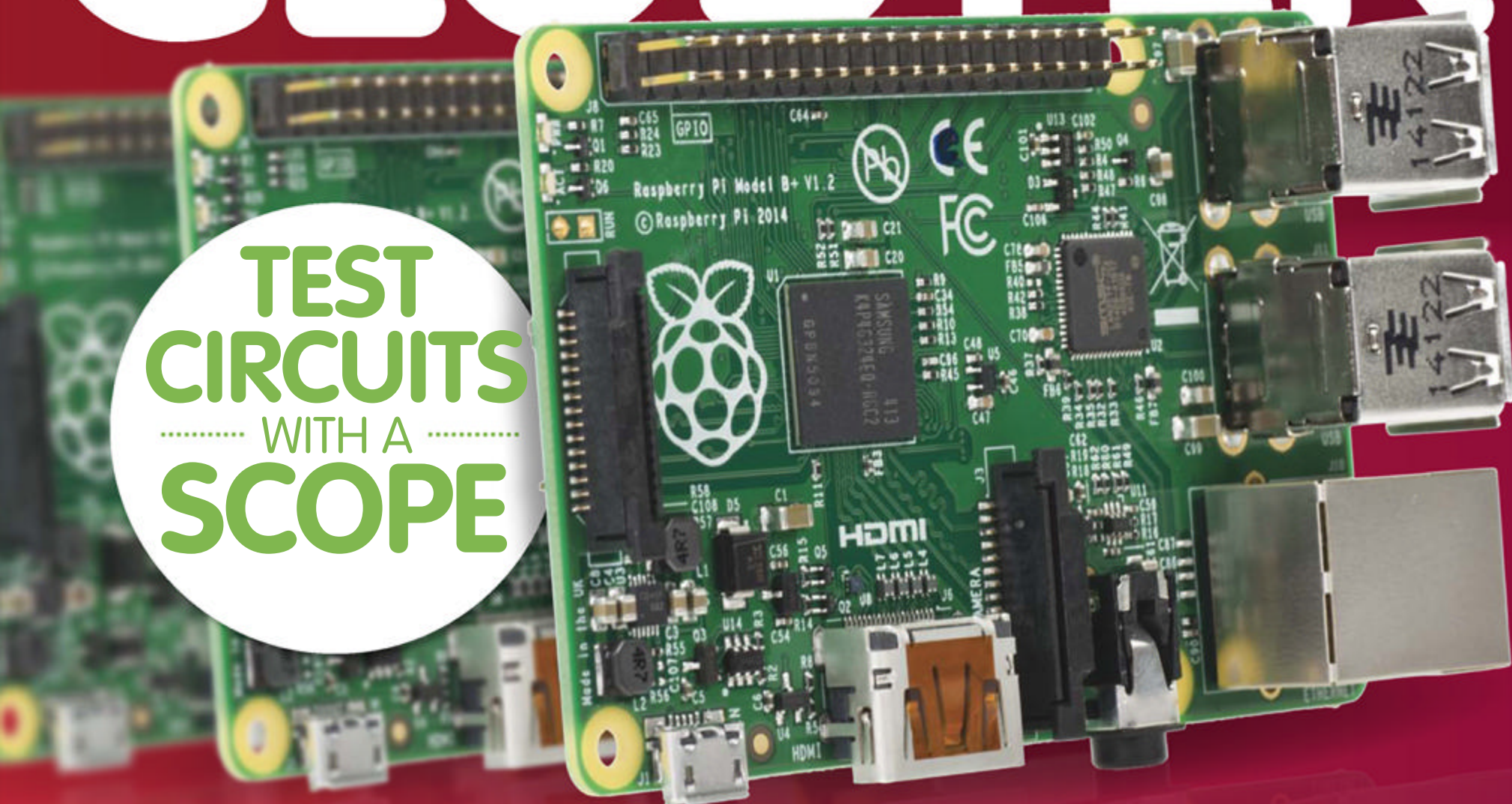
Get hands-on with your Raspberry Pi

Create a

RASPBERRY PI CLUSTER

TEST
CIRCUITS
WITH A
SCOPE

Plus Program Arduino on your Pi





Welcome



If you're anything like us, chances are that you've got a few spare Raspberry Pis knocking about that have

effectively been replaced by your shiny new Model 2B. That doesn't mean they have to stay in your drawer, though – it is a *lot* simpler than you think to simply chain them together and have them share their processing power in one big Raspberry Pi cluster. Get enough of them and you will basically have your own supercomputer. It's great fun to set up and opens up some incredible programming possibilities – just swipe forward a couple of pages to get cracking. And once you've got a cluster up and running, be sure to let us know what you're going to do with it – have fun!

Gavin Thomas

Editor

From the makers of
Linux User
& Developer

Join the conversation at...

 @linuxusermag

 Linux User & Developer

 RasPi@imagine-publishing.co.uk

Get inspired

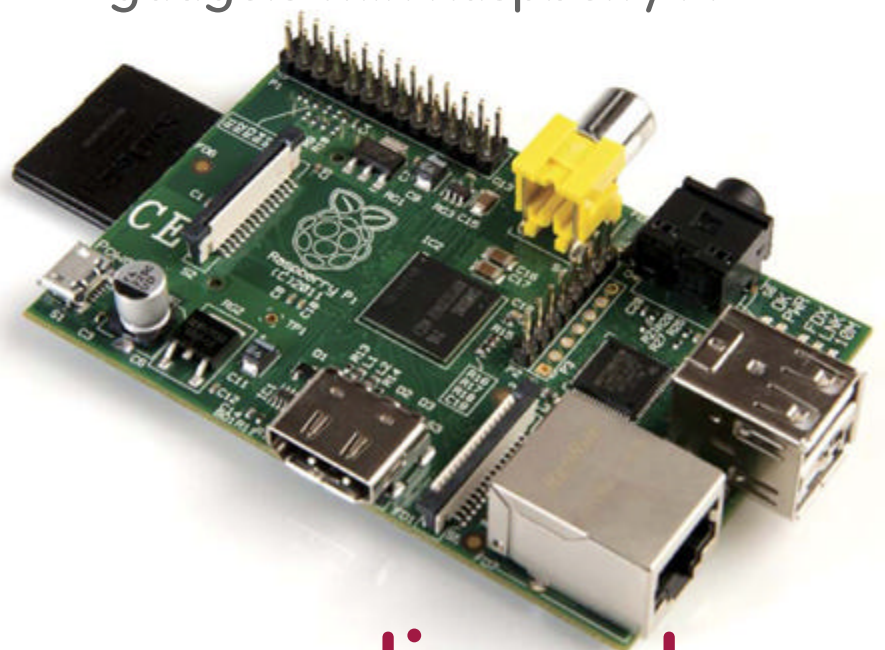
Discover the RasPi community's best projects

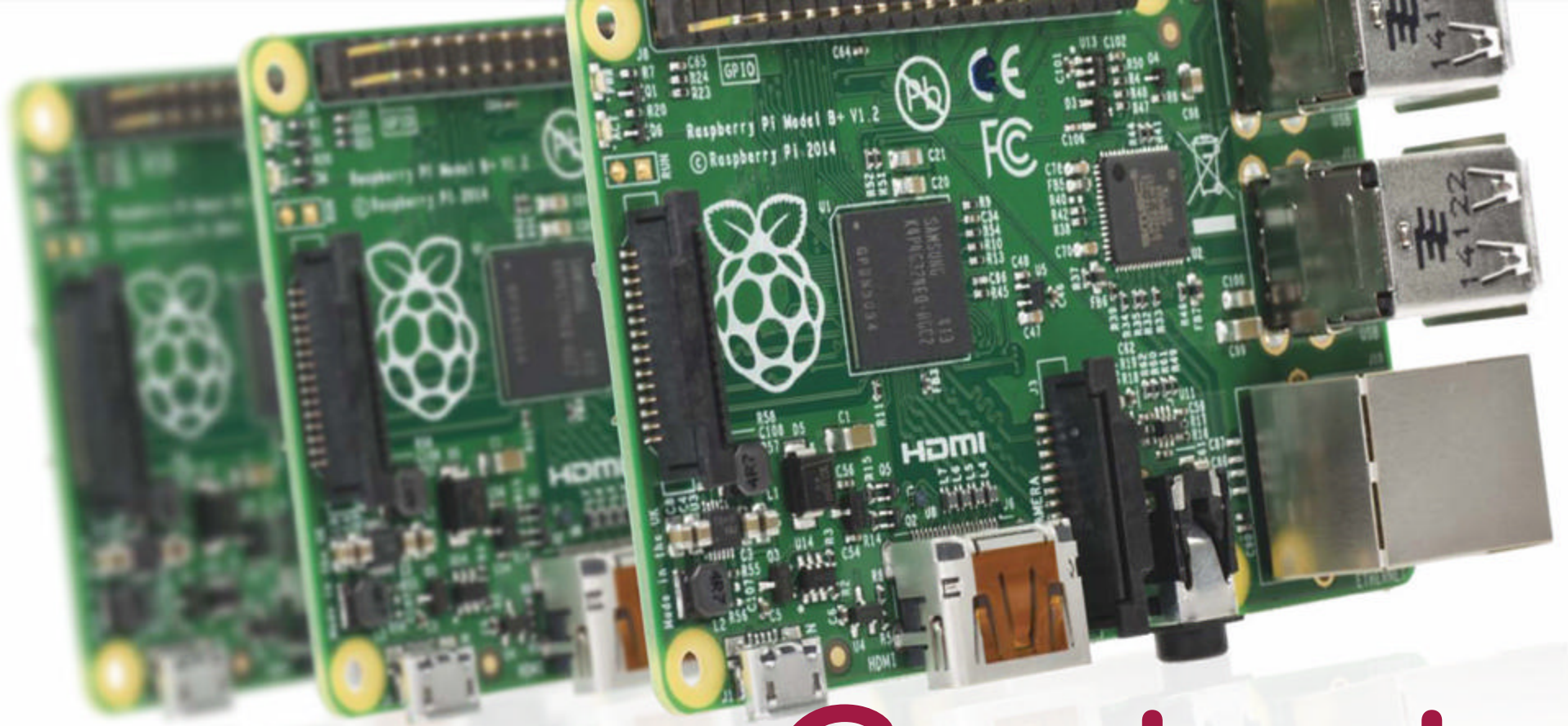
Expert advice

Got a question? Get in touch and we'll give you a hand

Easy-to-follow guides

Learn to make and code gadgets with Raspberry Pi





Contents

Build a Raspberry Pi cluster

... and then crunch code like a CERN scientist



Manage your cluster with IPython

Configure your Pis to handle parallel programming



What is Python?

We use it all the time but why is it so special?



Program Arduino on your Pi

Take advantage of the awesome Arduino IDE



Joytone

Check out the keyboard made of Xbox joysticks



Transform your Pi into a micro oscilloscope

Get to grips with the ultimate tool for electronics



Talking Pi

Your questions answered and your opinions shared

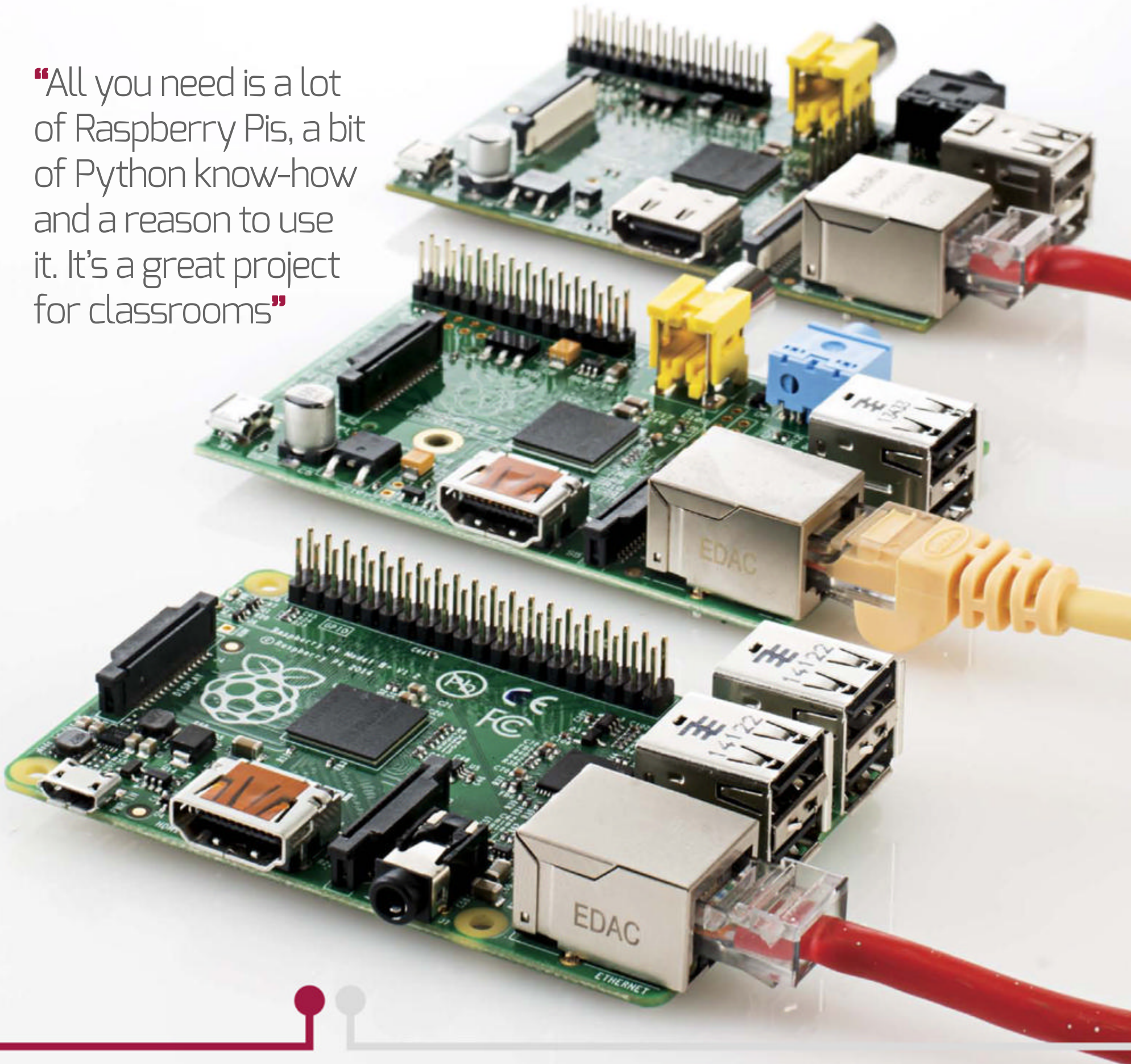




Build a Raspberry Pi cluster

Pool the resources of multiple Pis to create your own scalable Pi supercomputer

“All you need is a lot of Raspberry Pis, a bit of Python know-how and a reason to use it. It’s a great project for classrooms”





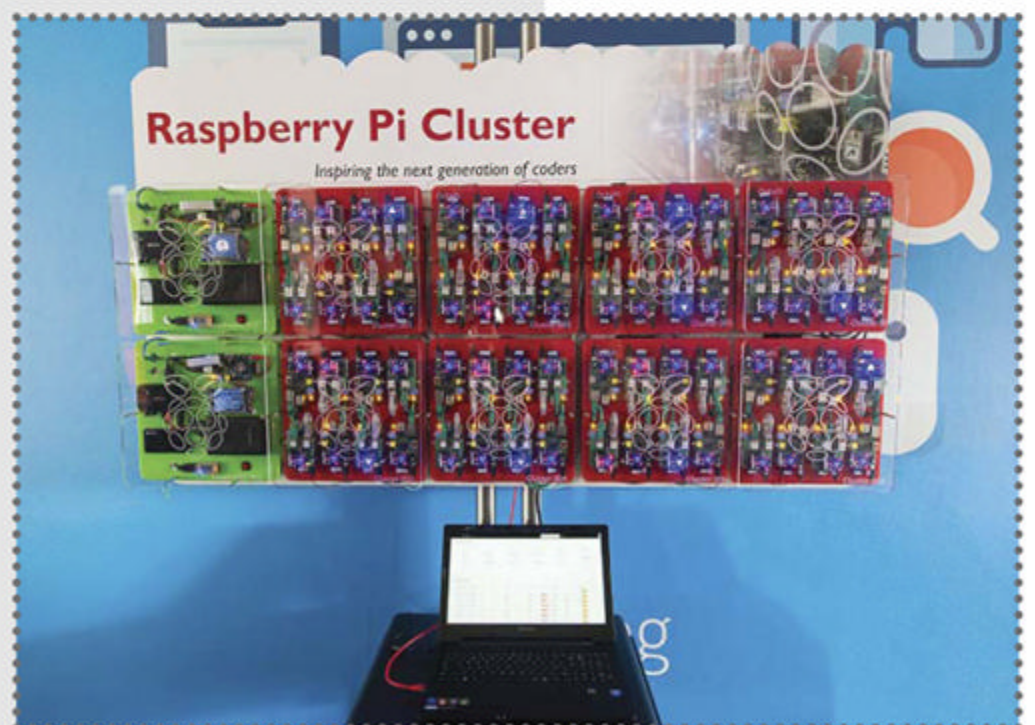
The Raspberry Pi is actually quite powerful for its price. On its own, though, you won't be doing any extraordinary calculations – or compiling, or anything strenuous at all, for that matter. However, as it's readily available and fairly cheap, you can get twenty of them for the price of a new computer. Each of them on their own will be no different, but link them together over a network and you can have them share their power and vastly increase the amount they can process.

This kind of setup is generally known as a Beowulf cluster, so named for the eponymous hero of the epic poem in which Beowulf is described as having “thirty men's heft of grasp in the gripe of his hand”. It's not ridiculously hard to achieve, either – all you need is a lot of Raspberry Pis, a bit of Python know-how and a reason to use it. This makes it a great project for things like classrooms, after-school Pi clubs and the like – really, anywhere there's a collection of Pis available for general use. The more Raspberry Pi nodes you add into the setup, the more powerful it will become, which means you can start with just two or three at home and then gradually add more and more to your cluster, if you want to. And because of the way it all works, you can hook in and control your Pi cluster from your main computer as well, making this as accessible at it is scalable.

Over the next few pages we're going to show you how to get your Raspberry Pis set up ready for use, including all the tools you'll need, how to get them all connected and then finally what you can do with all that processing power. We'd love to see your cluster once it's finished – drop us a tweet!

“You can hook in and control your Pi cluster from your main computer as well, making this as accessible at it is scalable”

Below GCHQ, the UK's NSA, have a cluster of 66 Raspberry Pis named the Bramble: <http://bit.ly/1wuTepj>





Programming your Pis

Get your Pis set up to talk to one other
and share their processing power

01 Fully updated Pi

It's very important that your Raspberry Pi computers are fully updated for this, so make sure everything is compatible, including the firmware. This can be done with three commands in succession, and make sure you do it on every Pi in turn:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo rpi-update
```

02 Get the MPI module

Install the mpi4py Python module in the terminal by using the following command (again, on every Pi):

```
$ sudo apt-get install
python-mpi4py
```

The module will also work on Arch or other Raspberry Pi distros, although you'll need to add it manually or install it via pip.

Right The Foundation can help you with using pip: <http://bit.ly/1L2Ubdb>



**THE PROJECT
ESSENTIALS**

**As many Raspberry Pi
computers as you want**

Latest Raspbian image

<https://www.raspberrypi.org/downloads/>

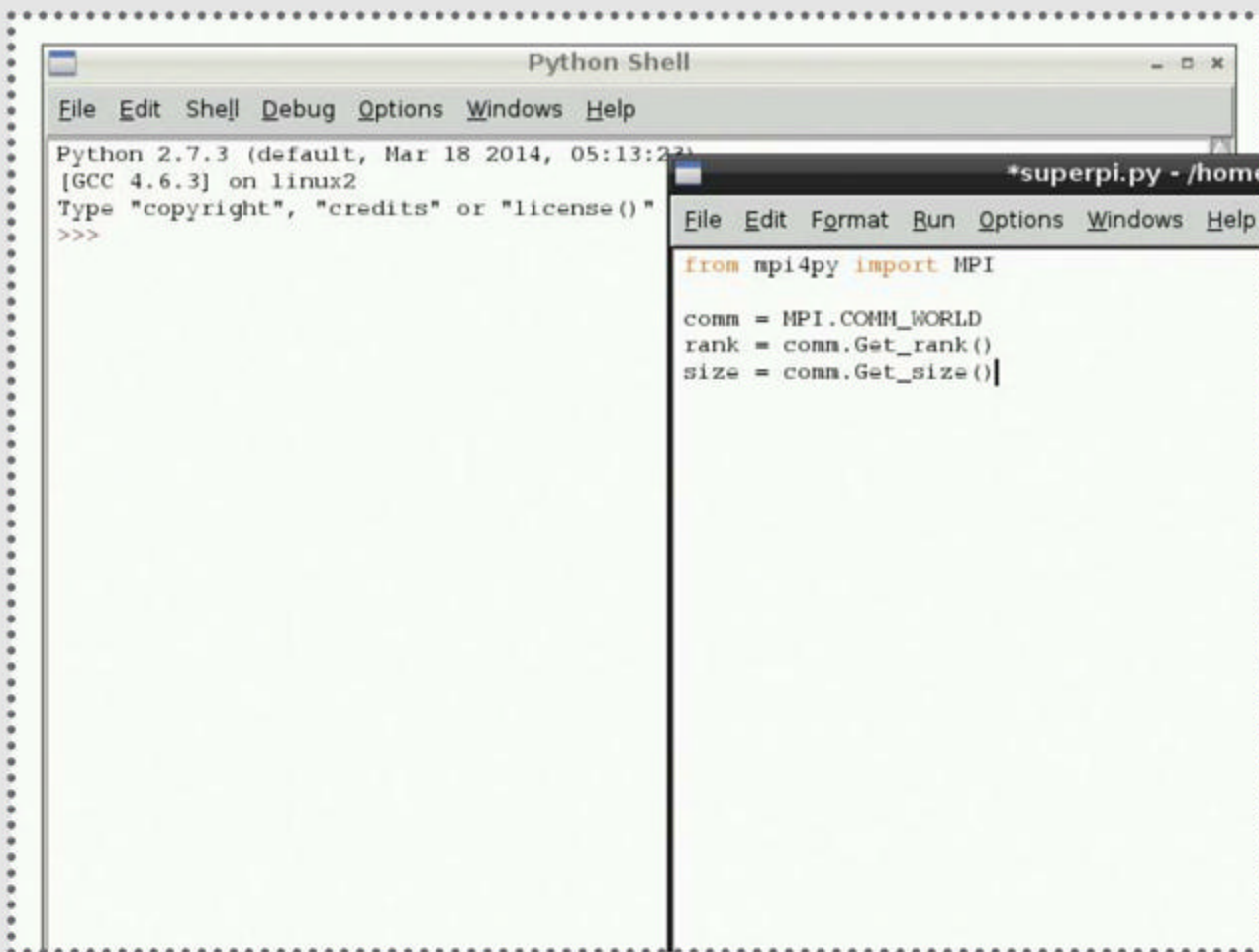
**mpi4py MPI python
module**

pypi.python.org/pypi/mpi4py

Updated firmware

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi ~ $ sudo apt-get install python-mpi4py
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libibverbs1 libopenmpi1.3 libtorque2 mpi-default-bin openmpi-bin openmpi-comm
Suggested packages:
  gfortran
The following NEW packages will be installed:
  libibverbs1 libopenmpi1.3 libtorque2 mpi-default-bin openmpi-bin openmpi-comm
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,485 kB of archives.
After this operation, 6,898 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libtorque2 armhf
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libibverbs1 armhf
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libopenmpi1.3 ar
20% [3 libopenmpi1.3 367 kB/1,206 kB 30%]
```





Left MPI stands for Message Passing Interface and is a communication system designed for parallel programming

03 Create your threads

To use MPI in Python, first import it with:

```
from mpi4py import MPI
```

The most important part of the code is telling MPI how to rank the threads and recognise their size. Do this with:

```
comm = MPI.COMM_WORLD  
rank = comm.Get_rank()  
size = comm.Get_size()
```

04 Send and receive data

Sending data is quite easy: you need to give it a destination thread so MPI knows what it is when it returns, halting the original thread until it does:

```
data = [1.0, 2.0, 3.0, 4.0]  
comm.send(data, dest=1, tag=0)
```

```
data = comm.recv(source=0, tag=0)
```

The tags allow you to parse what to do with some specific types of data.

Other libraries

The mpi4py library can also be used with Numpy, the numerical and mathematics module for Python. This can be installed using **sudo apt-get install python-numpy**, and is very useful for doing the kind of large calculations that a cluster would be making. We'll be using it in our example code in a couple of pages.

05 Run the code

So that's the basics of how your code can be sent throughout a cluster. To actually activate the code and choose the Raspberry Pi recipients, you will need to write this in a terminal:

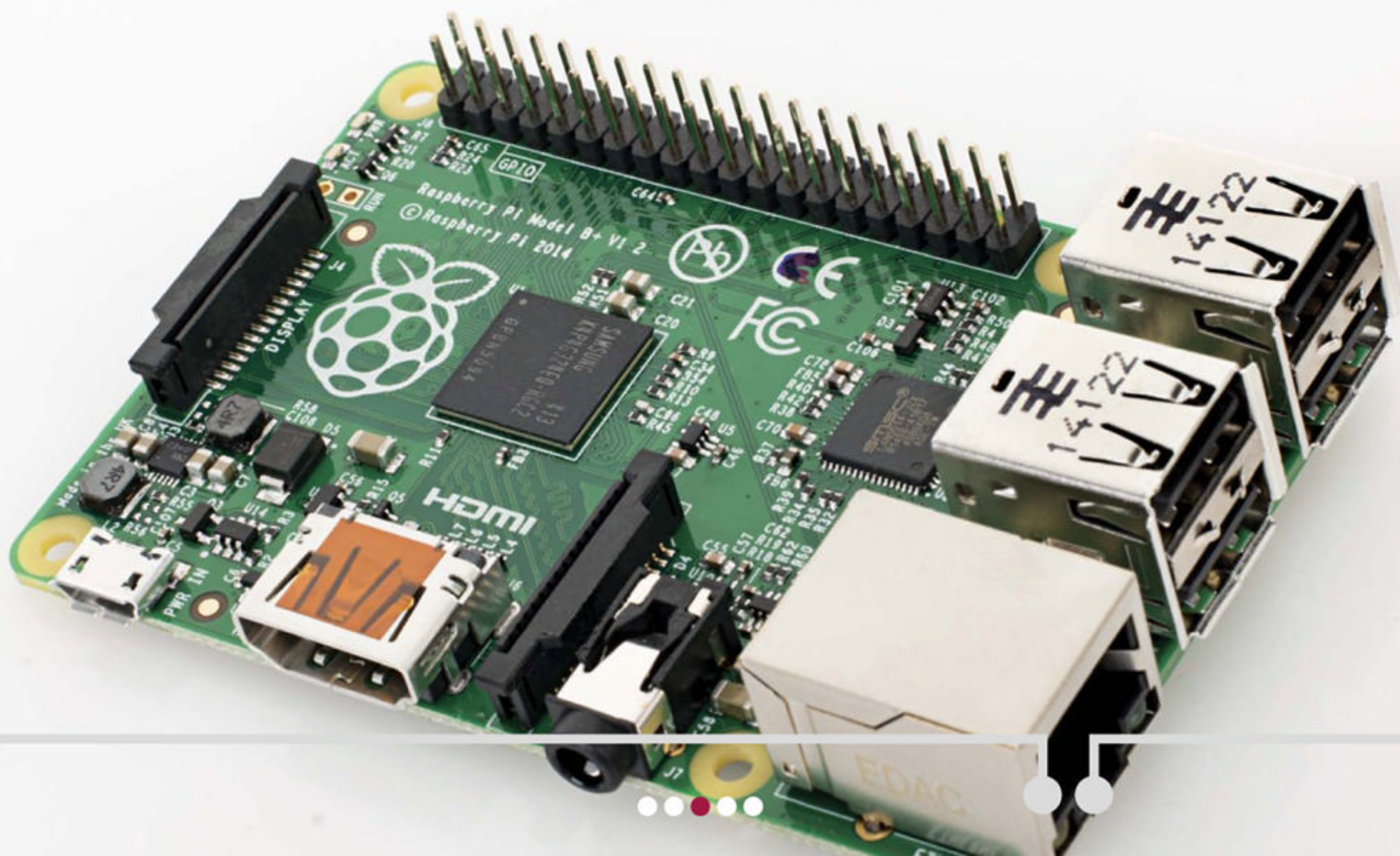
```
$ mpirun -host 192.168.1.20,192.168.1.21,192.168.1.22 python superpi.py
```

... with the IP addresses being the ones that are on the other Raspberry Pis.

06 Will it work?

What we've done above won't really do anything, but it at least illustrates how you can send code around the network. You can do it for any type of calculating but, due to some of the network lag, it's only worth it for large numbers and bigger data.

Below The biggest Pi cluster ever made so far has 120 nodes and is, appropriately, called The Beast: <http://bit.ly/1IRkkJJ>



The Code

CLUSTER TEST

```
from __future__ import division
```

```
import numpy as np
from mpi4py import MPI
```

```
# Grab parutils from FileSilo or our website and put it in the same folder as this.
Use it to then print the number of cores used
```

```
from parutils import pprint
```

```
comm = MPI.COMM_WORLD
```

```
pprint("--*78)
pprint(" Running on %d cores" % comm.size)
pprint("--*78)
```

```
# Set up a ranking structure for the different nodes
my_N = 4
N = my_N * comm.size
```

```
if comm.rank == 0:
    A = np.arange(N, dtype=np.float64)
else:
    A = np.empty(N, dtype=np.float64)
```

```
my_A = np.empty(my_N, dtype=np.float64)
```

```
# Scatter data into my_A arrays
comm.Scatter( [A, MPI.DOUBLE], [my_A, MPI.DOUBLE] )
```

```
pprint("After Scatter:")
for r in xrange(comm.size):
    if comm.rank == r:
```


The Code

CLUSTER TEST

```
print "[%d] %s" % (comm.rank, my_A)
comm.Barrier()

# Everybody is multiplying by 2

my_A *= 2

# Allgather data into A again and print results
comm.Allgather( [my_A, MPI.DOUBLE], [A, MPI.DOUBLE] )

pprint("After Allgather:")
for r in xrange(comm.size):
    if comm.rank == r:
        print "[%d] %s" % (comm.rank, A)
comm.Barrier()
```

“The most important part of the code is telling MPI how to rank the threads and recognise their size”



Construct your SuperPi

Now you've programmed your Raspberry Pi units, it's time to link them together



Programming the Raspberry Pis is one part of constructing your supercomputer – properly housing them together is another problem.

The very basic things you require to get them to work is power and a local network via a router and/or switches. This then extends to cases, SD cards and even Raspberry Pis themselves. To make sure you get the most out of the system, you'll have to get the right selection of components to power and connect them.

Power supply

A good power supply is extremely important when powering a Raspberry Pi, otherwise it will power on and run but you won't be able to get the most out of it. If you're using the Model B+ and notice a rainbow square in the corner, that's an indication you're not giving it enough power.

You won't need the maximum amount of power in your cluster, as you won't be powering a display or a camera or possibly anything off the GPIO ports, so the recommended two amps may be a

Below Pimoroni's power supply is a great buy at just £7: <http://bit.ly/1rkF5Gg>



little overpowered for your needs, even if you do push the processor to its limits. However, if you plan to do any overclocking then the two-amp power supply will become more and more necessary, and it's a lot better to be on the safe side.

As your cluster grows you'll also need to make sure you have plenty of surge protection as the Raspberry Pis can be extremely sensitive.

We recommend:
Raspberry Pi Universal Power Supply
pimoroni.com

“As your cluster grows you'll also need to make sure you have plenty of surge protection as the Raspberry Pis can be extremely sensitive”

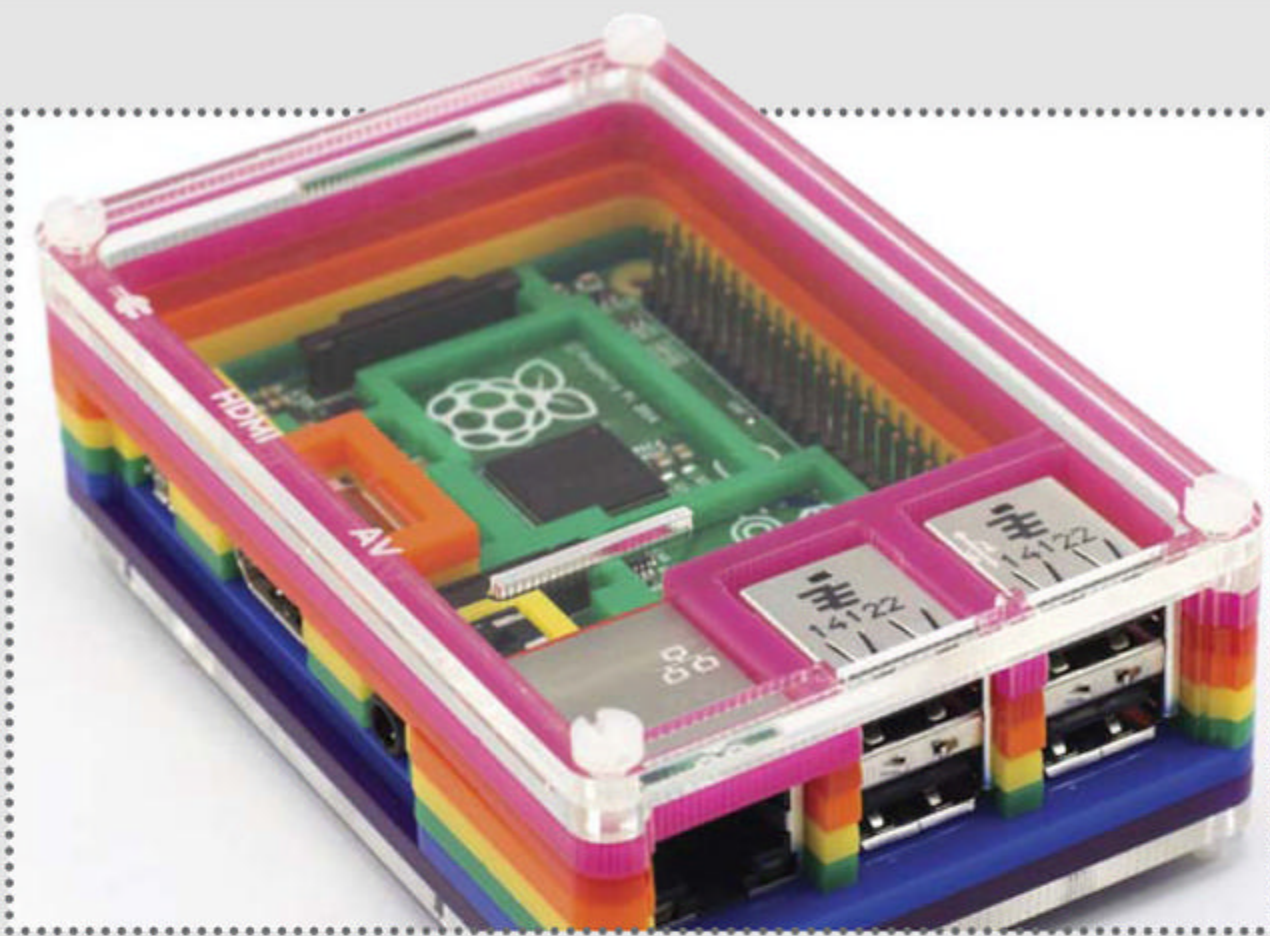
Networking

The more Raspberry Pis you plan to connect up, the more networking you will need to hook them together. The best method to do this is via cable, as it will carry any messages faster and with less latency than wireless – but creating the cluster isn't as easy as plugging it into a router, though. With more Pis you'll need more ports, and this can be accomplished by getting network switches. These are dumb connectors that contain no routing or address system, but allow you to connect more devices up to a central router.

Even though we recommend cabling, you can still use wireless to connect extra Pis. This also has the effect of cutting down on the amount of cabling and network switches you can use, but in general it's better to have a homogenous setup.

Cases

There are many types of cases for a lot of Raspberry Pi uses: from simplistic plastic cases to protect the Pi from



Left One of the oldest and greatest Pi cases around, the PiBow is a perfect, solid fit and has gaps for all your ports and cables

dust, to fully water- and weather-proof metal shells. One of the most versatile cases is the Pimoroni PiBow, which comes in a variety of sizes and configurations for all versions of the Raspberry Pi. The cases are all held together by screws that can easily be repurposed to help mount the Pis in a tower or other configuration so that they don't take up as much horizontal space.

They can be a little slower to get your Raspberry Pi into than some cases, but they're quite durable and easily customisable if you want to move them, or add a small touchscreen or scrolling LCD screen to the main Raspberry Pi to keep an eye on things without needing to SSH in.

We recommend:
Pimoroni PiBow
pimoroni.com

SD cards

SD card selection is a minor but still important factor when creating your cluster. Some Raspberry Pis come with an SD card which should be suitable enough, but others you'll

“The cases are all held together by screws that can easily be repurposed to help mount the Pis in a tower”

need to buy some cards for. We recommend getting 4GB cards; while a 2GB card will do the job, 4GB allows you to use NOOBS if you have to and is also future-proof for larger distros and operating systems. As the Model B+ requires a micro SD card, make sure you have the right ratio of SD and micro SD cards.

When it comes to getting the SD cards all set up and homogenous, the easiest and quickest method is to first do all the installing and updating on one Raspberry Pi, minus the firmware update with rpi-update. Then create a copy of the disk by putting it into a Linux system, open up the terminal and use:

```
sudo dd bs=1M if=/dev/[SD card location]  
of=superpi.img
```

Once it's copied, you can write this to all the other cards using:

```
sudo dd bs=1M if=superpi.img of=/dev/[SD card  
location]
```

We recommend:

4GB

pimoroni.com

Raspberry Pis

It actually doesn't really matter what type of Raspberry Pi you use in your cluster – you could use a homogenous selection of the latest Raspberry Pi 2 Model B or even have a mixture of the B+, B and Model As connected to each other. As long as you have them running the same software and possessing the relevant scripts, the system

“Do all the installing and updating on one Raspberry Pi, minus the firmware update with rpi-update. Then create a copy of the disk”

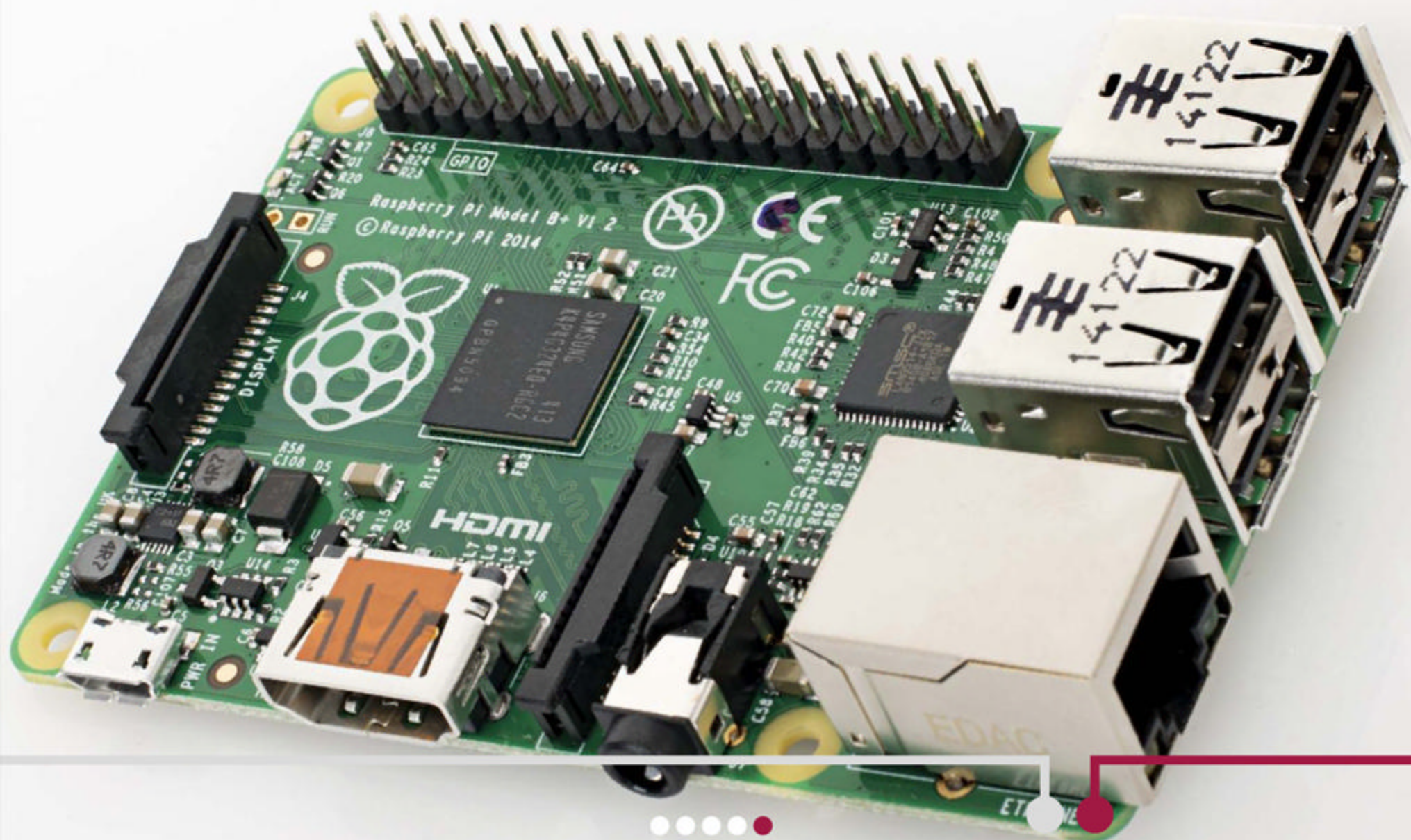
will work. The main differences you might encounter are the differing power draws between devices and that the Model As might be slightly slower for some calculations.

The Model A types do have a disadvantage in that they do not have an ethernet port built-in. However, they can still make use of a wireless dongle to connect to the overall network of Raspberry Pis.

Make sure you set a static IP address for each Raspberry Pi with a specific range covering their location on the network. This is helpful for two main reasons, the first one being the ability to always be able to call the correct address when running MPI, and the second being that you can then SSH in and maintain your Pis from afar.

We recommend:
Model B+
element14.com

Below If you've got a shiny new Pi 2, your B+ is now the perfect candidate for a Beowulf cluster





How to use your cluster

Your supercomputer is ready to crunch some code. What will you have it do?



Donate power with BOINC

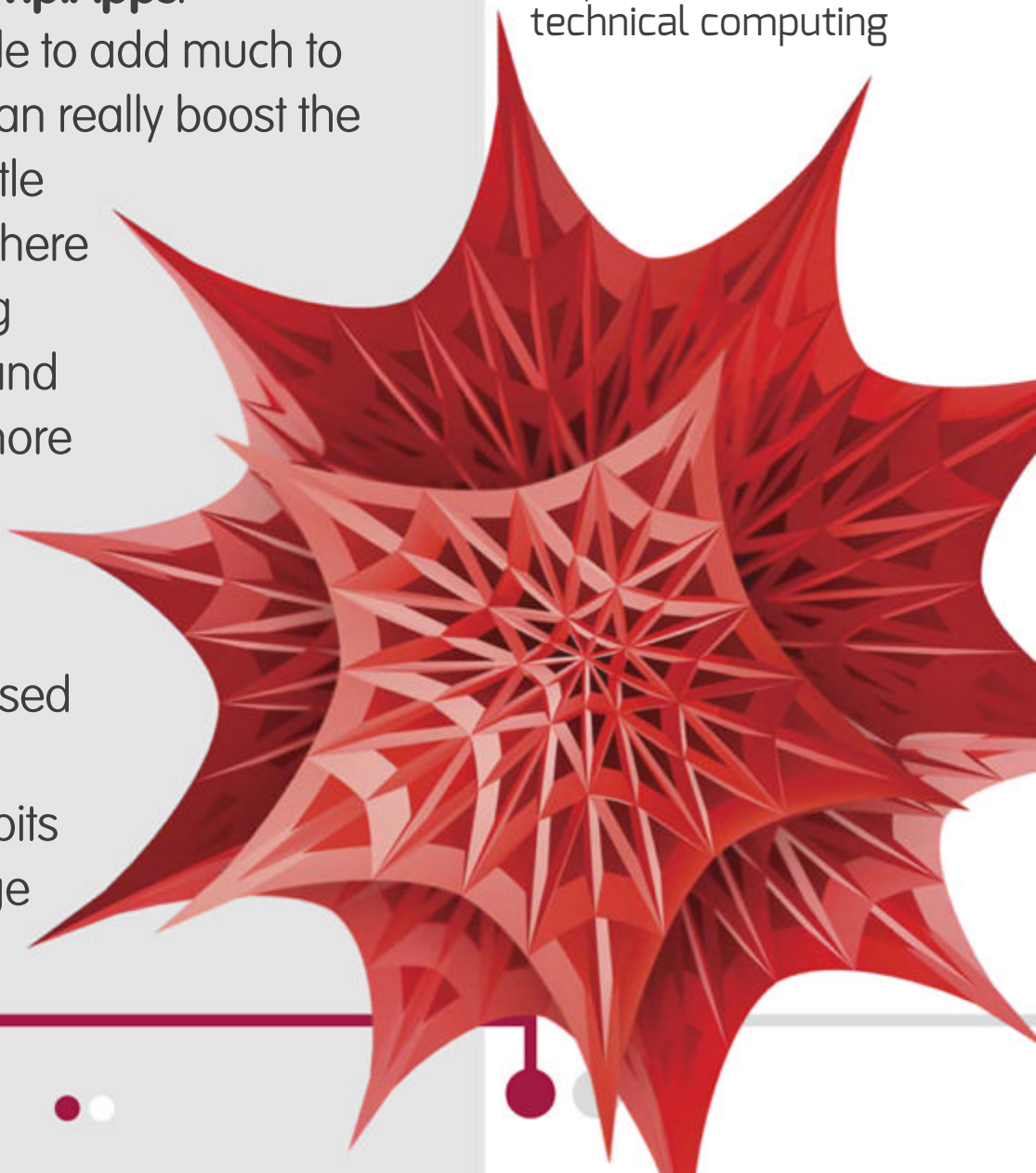
BOINC is the computer network system that allows you to donate idle CPU power to crunch big data, such as folding proteins or analysing signals from the cosmos to look for alien life. There's a specific app add-on for BOINC that lets you spread the workload across the MPI network and treat them all as extra CPU threads rather than extra units. The application details can be found here: <http://boinc.berkeley.edu/trac/wiki/MpiApps>.

The Ras Pi on its own won't be able to add much to the pool, but linking several together can really boost the performance and create a powerful little node to help out with some projects. There are several you can donate processing power to involving science, medicine and maths. You can tweak BOINC to use more than just idling Pi power.

Modelling and simulations

Python and other languages can be used to create models of complex systems. Such models can include planetary orbits and objects interacting with them, huge

Below Wolfram's Mathematica is a great way for you to explore the power of your cluster for technical computing



mathematical equations and any resulting graphs for tides, the weather and much more. Many of these use a lot of calculations and can be slow even on a normal PC. Give them a lot more cores and you can get much faster, and sometimes more accurate, results.

Mathematica also contains some level of cluster support that works via Open MPI, the software we're using to link the Raspberry Pi network together. Wolfram has information on how it works and how to use it on their website (<http://bit.ly/1oEYVb8>), and it can really help with crunching the amount of data you can access and use inside Mathematica.

Quick compiling

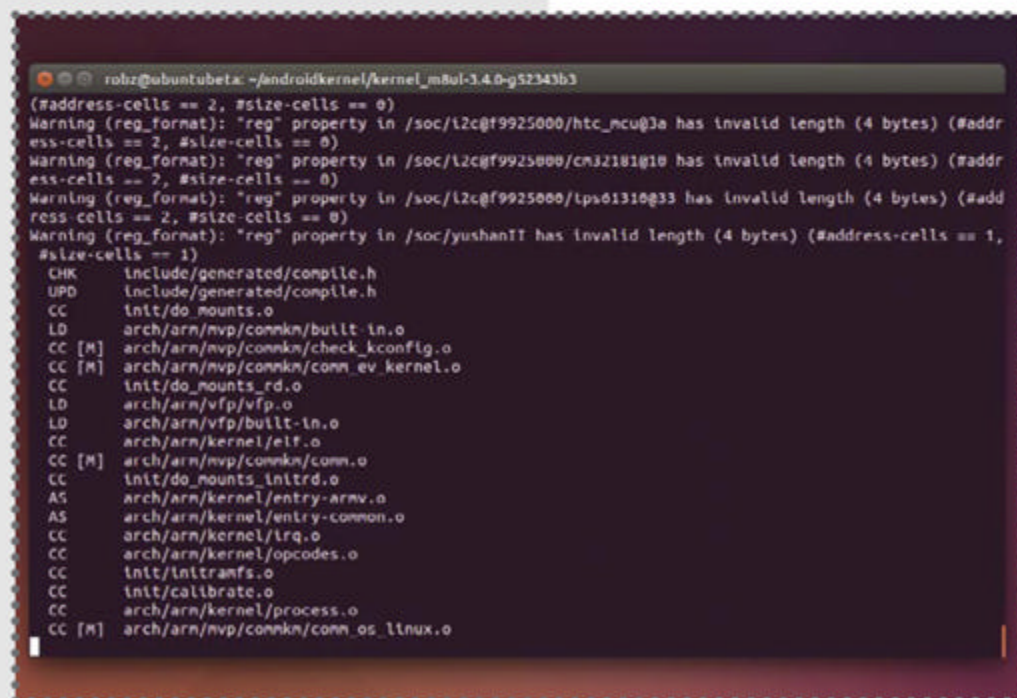
Now you have a lot more processing power at your disposal, you can try compiling Raspberry Pi programs on the cluster rather than just on the single Raspberry Pi. These will only really work on another Raspberry Pi, though, and compiling on a single Raspberry Pi can be extremely time-consuming compared to compiling on a proper PC or laptop.

There is, however, also a selection of MPI-only applications that can be built using your cluster to increase the functionality of cluster without relying on Python code.

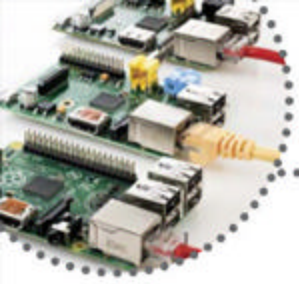
These must include specific MPI commands in the source code to make sure they run properly, which can be done a little differently than in our Python example. Look up the best practices for your programming language and if there are any extra modules or plugins you'll need for it. You can test the code in a similar way to how we tested the Python code.

“Models can include planetary orbits and objects interacting with them, huge mathematical equations and any resulting graphs for tides, the weather and much more”

Below By splitting the workload across your cluster, you can compile programs far quicker than before



```
robz@ubuntu: ~/androidkernel/kernel_m8ul-3.4.0-g52343b3
(#address-cells == 2, #size-cells == 0)
Warning (reg_format): "reg" property in /soc/l2c@f9925000/htc_mcu@3a has invalid length (4 bytes) (#address-cells == 2, #size-cells == 0)
Warning (reg_format): "reg" property in /soc/l2c@f9925000/cm32181@10 has invalid length (4 bytes) (#address-cells == 2, #size-cells == 0)
Warning (reg_format): "reg" property in /soc/l2c@f9925000/lps01310@33 has invalid length (4 bytes) (#address-cells == 2, #size-cells == 0)
Warning (reg_format): "reg" property in /soc/yushan11 has invalid length (4 bytes) (#address-cells == 1, #size-cells == 1)
CHK include/generated/compile.h
UPD include/generated/compile.h
CC init/do_mounts.o
LD arch/arm/mvp/connkn/built-in.o
CC [M] arch/arm/mvp/connkn/check_kconfig.o
CC [M] arch/arm/mvp/connkn/conn_ev_kernel.o
CC init/do_mounts_rd.o
LD arch/arm/vfp/vfp.o
LD arch/arm/vfp/built-in.o
CC arch/arm/kernel/elf.o
CC [M] arch/arm/mvp/connkn/conn.o
CC init/do_mounts_initrd.o
AS arch/arm/kernel/entry-armv.o
AS arch/arm/kernel/entry-common.o
CC arch/arm/kernel/irq.o
CC arch/arm/kernel/opcodes.o
CC init/initramfs.o
CC init/calibrate.o
CC arch/arm/kernel/process.o
CC [M] arch/arm/mvp/connkn/conn_os_linux.o
```



Resources

Find out more about MPI, Python and other ways to make use of your Beowulf cluster



Open MPI <http://open-mpi.org>

Open MPI (right) is the implementation of MPI that our Python module uses to send and receive data over the Raspberry Pi network. It supports all three major versions of MPI and, importantly, provides it in an open source format for everyone to use. There's full documentation for the extended library on the forums (www.mpi-forum.org) if you need to extend it past Python programs and take a little more control of what you're calculating and how.

Below The official Raspberry Pi forums host an ever-growing community of Pi enthusiasts –help is always at hand, no matter what you're working on

Raspberry Pi forums

www.raspberrypi.org/forums

For anything going wrong with your Pi beyond MPI, your first port of call should always be the Raspberry Pi forums. The users are well experienced in using the Raspberry Pi and Linux, and will usually be able to help out. If you can give precise details and log files along with your post, they might be able to help you quicker.

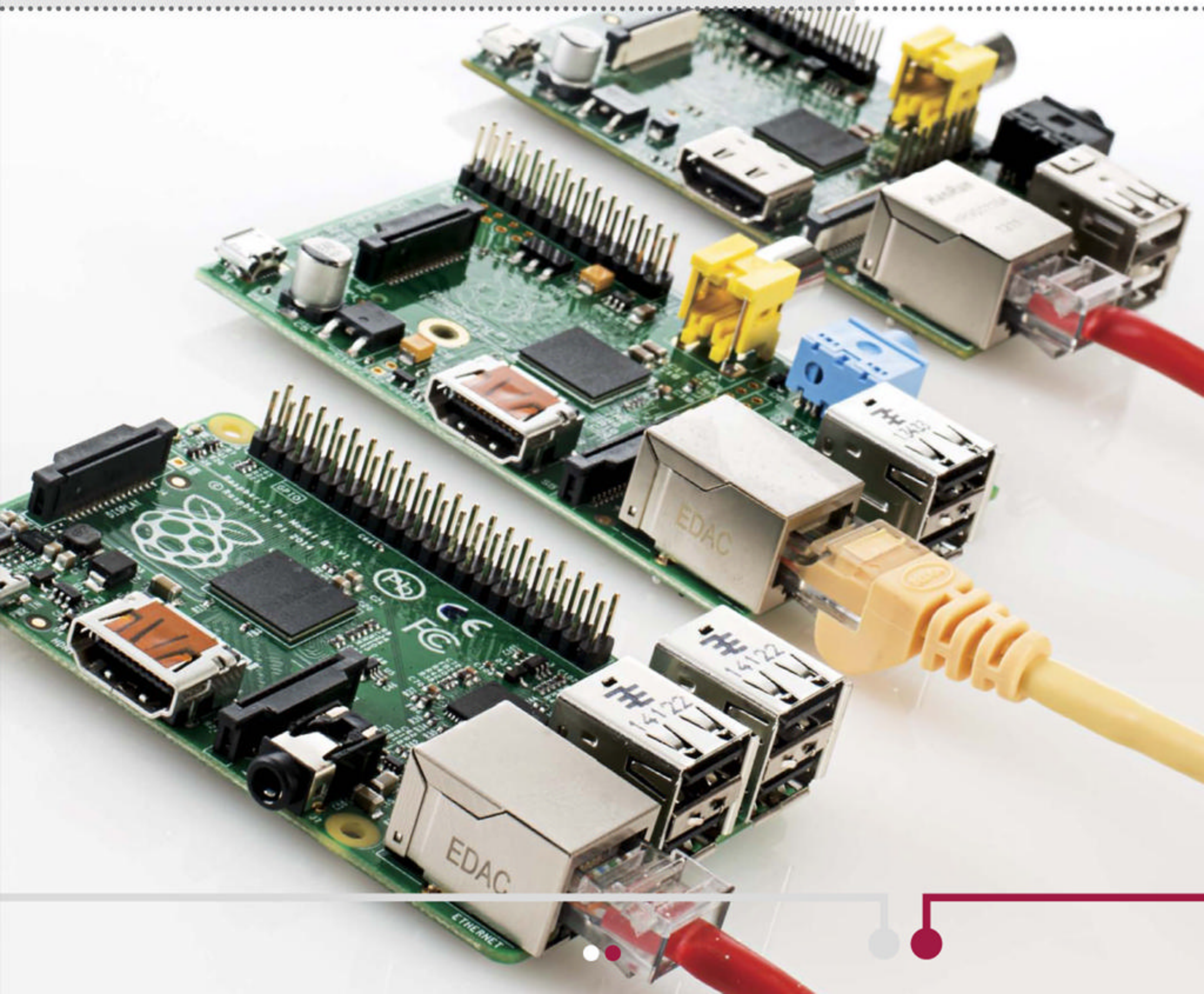
Raspberry Pi®					Quick Start	Downloads	Buy Codecs	Forum	FAQs	About
FAQ Register Login					Search...					
Community										
Forum	Topics	Posts	Last post	When						
General discussion	13205	123949	by littenkyling	Wed Aug 06, 2014 6:36 pm						
Other languages	8079	39944	by hoxpott	Wed Aug 06, 2014 6:03 pm						
User groups and events	446	2151	by FlapBumWale	Tue Aug 05, 2014 2:08 pm						
The Pi Store	152	976	by wlatstella	Sun Aug 03, 2014 11:31 pm						
Moderators: DrPiG, Pankol										
Using the Raspberry Pi										
Forum	Topics	Posts	Last post	When						
Beginners	9768	58419	by YurkoPhelcom	Wed Aug 06, 2014 6:02 pm						
Troubleshooting	10152	82780	by dgeoff	Wed Aug 06, 2014 6:02 pm						
Advanced users	1887	12390	by AndrewB	Wed Aug 06, 2014 6:08 pm						
Education										
Forum	Topics	Posts	Last post	When						
The Staffroom	310	8998	by Jim JGle	Wed Aug 06, 2014 11:38 am						

mpi4py documentation

<http://pythonhosted.org//mpi4py/usrman/index.html>

The Python module we're using has many more functions than the one we're using, and they're all contained in the documentation available on their website. You can learn how to do much more than just multiply numbers, as we showed you in our example, including the use of wrappers for other code, matrices of data and a couple other functions. It's very flexible and still in development so it's worth checking in on it and the change logs every now and then for new or different functions.

“It's very flexible and still in development so it's worth checking in on it and the change logs every now and then”





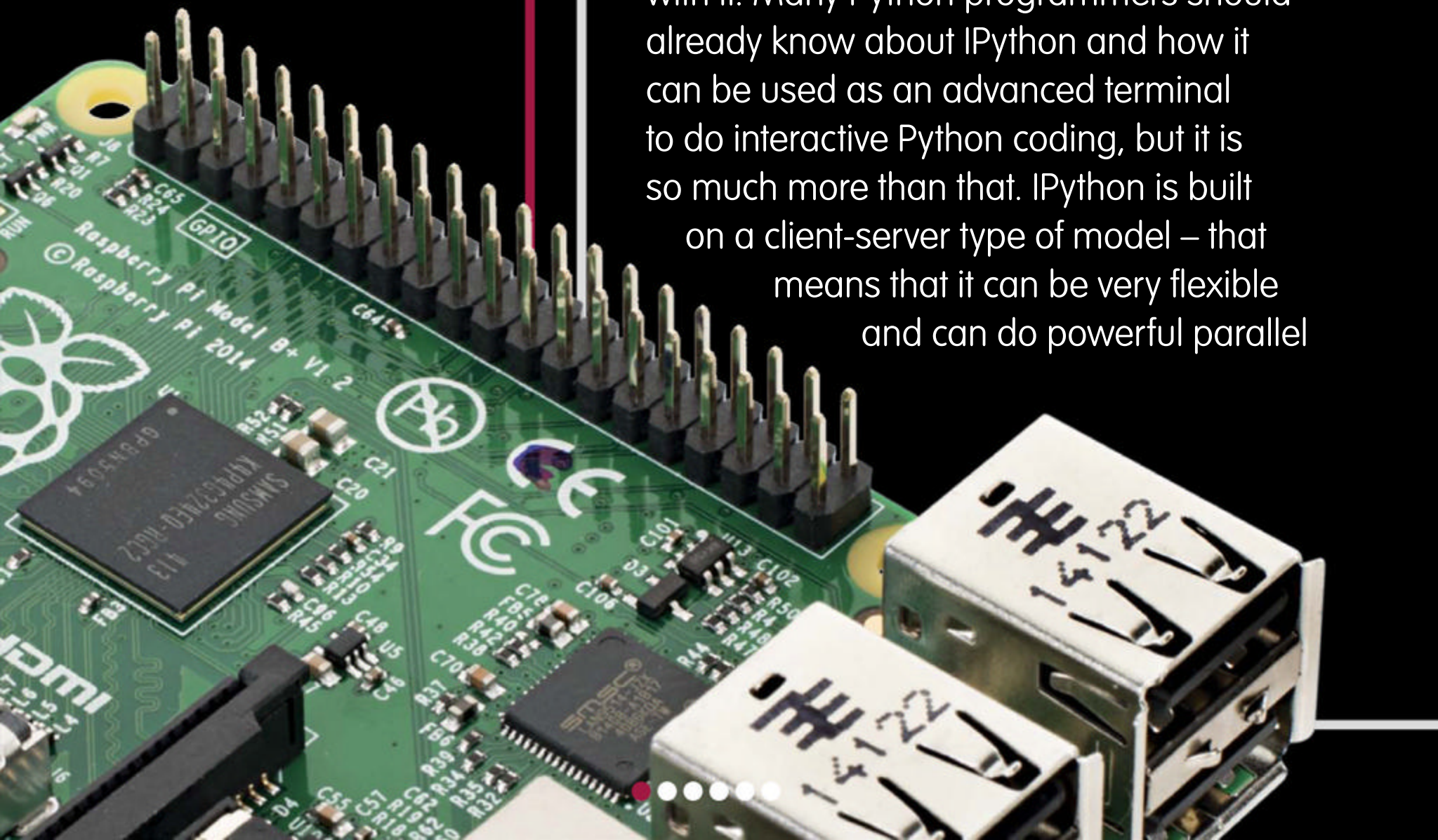
Manage your Raspberry Pi cluster with IPython

Learn how to configure a cluster of Raspberry Pis to handle parallel programming

“Each individual engine can only execute a single command at a time”



So, we have just taken a look at setting up and using a cluster of Raspberry Pis and then doing parallel computations with MPI. But with Python, there is more than one option available. Now we will look at IPython and see what kind of parallel work you can do with it. Many Python programmers should already know about IPython and how it can be used as an advanced terminal to do interactive Python coding, but it is so much more than that. IPython is built on a client-server type of model – that means that it can be very flexible and can do powerful parallel



programming. IPython supports lots of different parallel methodologies, including single program multiple data (SIMD), multiple program multiple data (MIMD), task farming, data parallelism and several other paradigms. With the flexibility afforded by the underlying structure, you can develop almost any type of parallel program. IPython is broken down into four separate sections: IPython engine, IPython hub, IPython schedulers and a controller client. To use IPython on the Raspberry Pi, you need to install the relevant packages. Assuming that you are using Raspbian, or something similar, you can install the required packages with the command:

```
sudo apt-get install ipython
```

One prerequisite that does not get installed automatically is zmq. You will need to install this manually with:

```
sudo apt-get install python-zmq
```

So, what can you do once you have IPython installed? We should take a quick look at how IPython is structured in order to get a better feel for how you can use it to do some really cool stuff.

The first portion to look at is the engine. An IPython engine is a Python instance that listens on a network interface and executes incoming Python commands. Incoming commands and outgoing results might include Python objects, too. When you start up multiple engines, you essentially have a parallel programming system at your disposal. One thing to be aware of is that each individual engine can only execute a single command at a time and is blocking while any particular command is being run. The solution to this potential

problem is the controller. A controller is made up of a hub and a collection of schedulers. This controller accepts connections from both engines and clients and acts as a bridge between the two sections. Users talk to the controller, and the controller communicates to the attached engines. This is done through a Client object that connects to a controller and hence to a cluster of CPUs. In order to do parallel programming, you need to create a cluster of IPython engines, and then you need to connect to it. To start up a cluster of engines, you need to use the command `ipcluster`. So, if you wanted to start up two engines on a given Raspberry Pi, you would use the following command:

```
ipcluster start -n 2
```

To test that everything is working, you could start the IPython interface and try to connect to these two new engines. The code would look like:

```
from IPython.parallel import Client  
c = Client()
```

You can check the number of engines available by querying the `ids` property of the newly created client object. In this case, you should see the list `[0,1]`. This might be okay if you are just testing some code out, but the whole point is to chain a number of Pis together as one and use them as slaves. We will start by assuming that you will be using a laptop to do your programming on. This machine will act as the frontend to your Raspberry Pi cluster. We are also going to assume that you are using

“A controller is made up of a hub and a collection of schedulers. This controller accepts connections from both engines and clients and acts as a bridge between the two sections”

some distribution of Linux. You will want to install the IPython and python-zmq packages on your laptop using your distribution's package manager. Once that is done, you will need to create a new IPython profile to store the configuration for your cluster. You can do this with the following command:

```
ipython profile create --parallel  
--profile=rpi
```

This will create a new directory named 'profile_rpi' in your IPython configuration directory and will vary depending on your distribution. In this directory, you will want to edit the file 'ipcluster_config.py' to set up the details for your cluster. In the sample code here, we are using SSH as the communication method. You can then set the hosts as a list, which is stored in the property 'SSHEngineSetLauncher.engines' of the configuration. The hosts are enumerated with the format 'hostname : number_of_engines'. In the sample code, we used IP addresses, since everything sits on an internal network. The other property you need to set is the command used to start an engine on the individual cluster nodes, which is stored in the property 'SSHEngineSetLauncher.engine_cmd'. If you are using Raspbian, this should simply be 'ipengine'. The last step is to be sure that the profile directories exist on the cluster nodes. On each Pi, you will want to run the command:

```
mkdir -p .ipython/profile_rpi/security
```

...since it doesn't get created automatically. You also need to be sure that all of the machines on the network can talk to each other cleanly. You may want to look into setting

up passwordless SSH, otherwise you will need to enter passwords when you try and start the cluster. These types of communication issues will likely be the primary cause of any issues you may have in getting everything set up. Now, you have an IPython cluster ready to run. On your local laptop, you can start the cluster up by running the following command:

```
ipcluster start --profile=rpi
```

A series of messages will display in the console. Once the controller has finished initialising, you can start using it. You simply create a new client object using the rpi profile and use it to do different types of parallel programming. You should now be able to start using all of those Raspberry Pis that you have been collecting. With IPython, you can rein them all in and get them working together on all of your largest problems. You will now have the tools to build one of the lowest-energy supercomputers available in the world.

“You will now have the tools to build one of the lowest-energy supercomputers available in the world”



The Code

IPYTHON CLUSTER

Edit the cluster configuration file `ipcluster_config.py`
Make sure the following lines exist in this file:

```
c = get_config()
c.IPClusterEngines.engine_launcher_class = 'SSH'
c.LocalControllerLauncher.controller_args = ["--ip='*']
c.SSHEngineSetLauncher.engines = {
    'localhost' : 2,
    'rpi1' : 1,
    'rpi2' : 1
}
c.SSHEngineSetLauncher.engine_cmd = ['ipengine']
```

In IPython, work with the engines you started up:

```
from IPython.parallel import Client

# Create a client and view for the cluster
my_client = Client()
my_view = my_client[:]

# You can map a function across the entire view
par_result = my_view.map_sync(lambda x: x**10, range(32))

# You can create a remote function that runs our
# on the engines
@my_view.remote(block=True)
def getpid():
    import os
    return os.getpid()

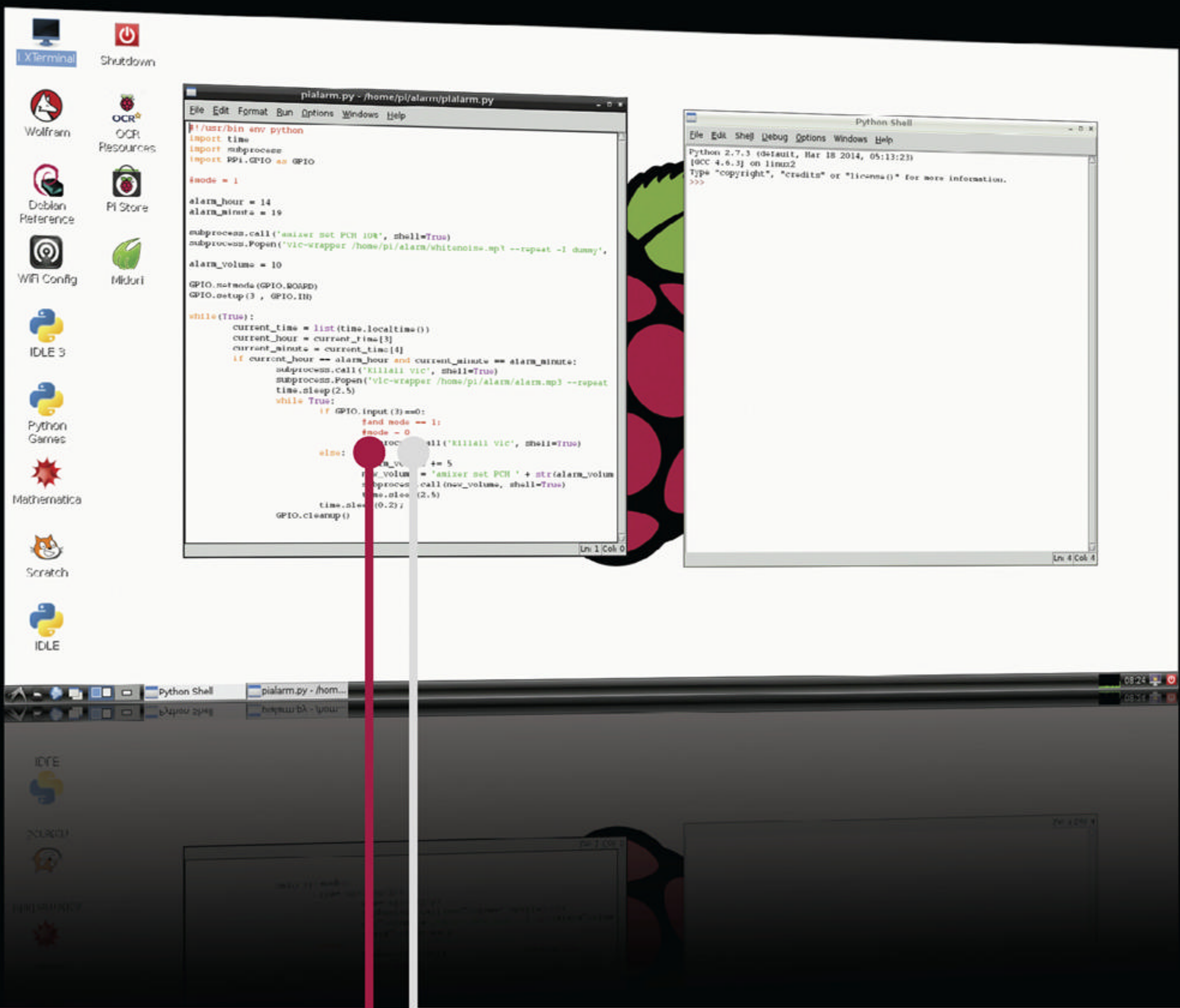
# Calling 'getpid()' will get the PID from each
# remote engine
```

“IPython supports lots of different parallel methodologies, including single program multiple data (SIMD), multiple program multiple data (MIMD), task farming, data parallelism and several other paradigms”



What is Python?

The popular programming language used on the Raspberry Pi. What makes it so special?



Q There's a lot of focus on Python for programming on the Raspberry Pi. Is this because it's the only way to program the Raspberry Pi?

A No, not at all. As we're generally using Linux on the Raspberry Pi, just about every major programming language can be used. C, C++, Ruby, Perl and more are all completely compatible with the Raspberry Pi. You're not really programming the Raspberry Pi with them either – you're creating programs that will run on the Raspberry Pi.

Q That's good to know, then. If that's the case, why concentrate on Python as much as you guys and the Raspberry Pi folk do?

A It really all comes back to the core mission of the Raspberry Pi Foundation: to help educate school children in computer science. This involves coding, and Python is considered an easy language to learn. It's a language a lot of people know anyway. Plus, we have a bit more expertise in Python on the magazine, and we're also very happy to promote the Raspberry Pi Foundation's objectives.

Q What makes it so easy to use over the other programming languages?

A Python is 'readable'. Core variables and functions are named plainly, the structure is designed to handle a lot of white space and everything is written a lot more straightforwardly than other programming languages. What it all means is you can quickly scan some Python code and make more sense of it than other languages, which can sometimes look a bit like a cat took a walk across a keyboard.



Q So readable means anyone can just look at it and understand what's going on?

A That's not quite what it means; you would still need a basic understanding of coding and Python to recognise what specific parts mean. These can include the difference between a tuple or a list or a class and a function. These can easily be learned, though, and are generally easier to pick up than with the other languages.

Q It's easy enough to teach to children?

A Absolutely. It's very similar to maths in that you're teaching rules and methods that you can immediately produce an answer or output from. With the correct lessons, different concepts can be introduced and built upon just like anything taught in school. In fact, the incoming changes to the UK curriculum will cover coding in this exact way.

Q If Python's such a simple language, does this mean it's not very useful in the real world of programming?

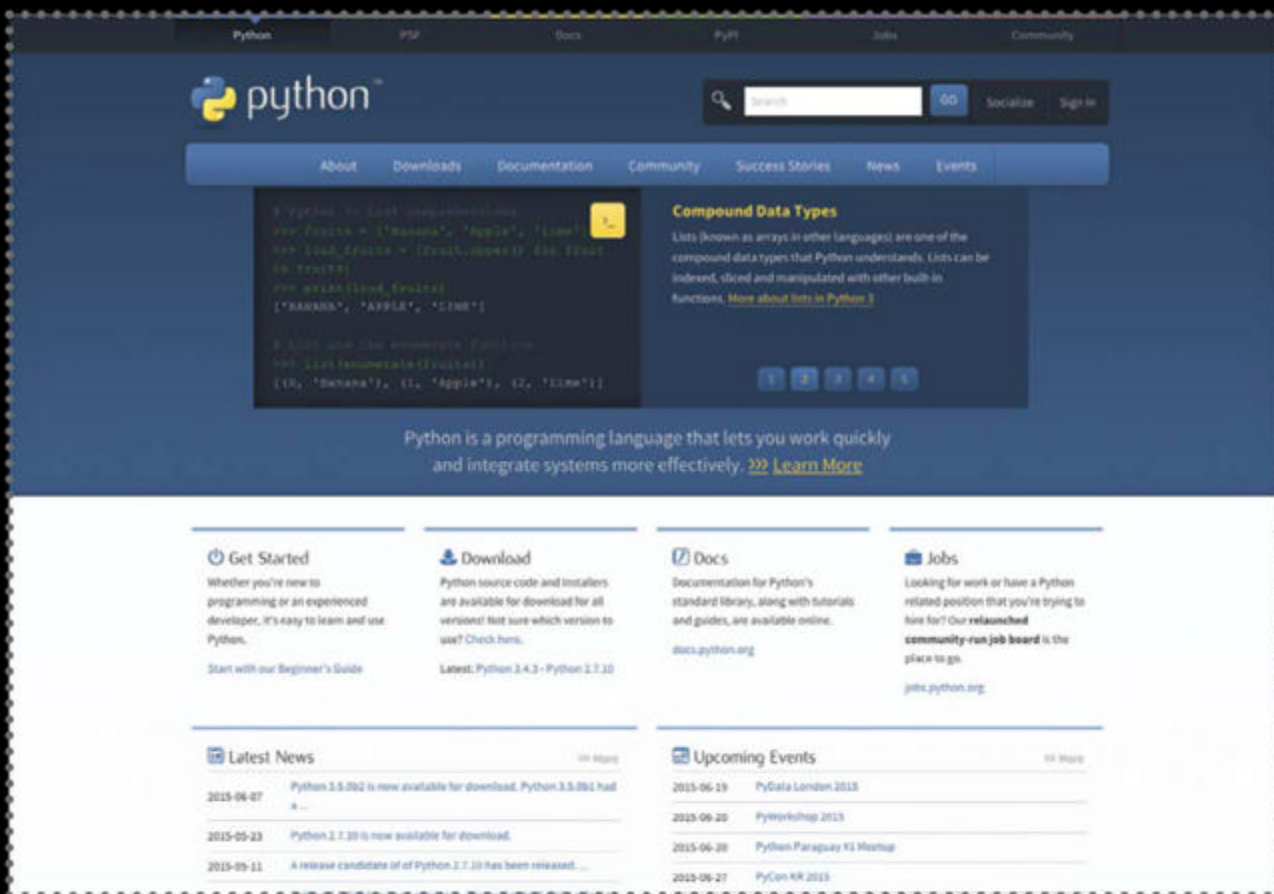
A That's not the case at all – Python is used by companies around the world in 'real programming'. It's not the only language, though, and some companies won't even use Python – at the very least, it sets people up to learn the intricacies of programming languages and makes learning others a little bit easier.

Q Can Python interact with other programming languages?

A It largely depends on the language but the quick answer is not really. There are very few, if any, projects where you'll need to use different languages, unless they handle completely different aspects of the software. A database could be created and maintained using SQL but that's something Python can't handle itself anyway.

“It's similar to maths in that you're teaching rules and methods that you can immediately produce an answer or output from”





Left The Python Software Foundation have an excellent website with useful guides to Python: www.python.org

Q How do I create a Python program?

A Like a lot of programming, you need to create a script: a file that contains all your code and tells a Python interpreter what to do when the code is run properly. You can create them in a plain text editor like gedit or even nano on the terminal and save them as .py files before testing them. The best way is to create it in an IDE though.

Q What is an IDE? How do I get one on the Raspberry Pi?

A An integrated development environment is software that lets you create and test scripts in specific languages. You usually have a few more ways to debug your files with them as well. In the case of Python on the Raspberry Pi, you can use IDLE which is already installed and available from the desktop.

Q Wait... IDLE? Python? Is that an intentional reference?

A Yep, IDLE is named after Monty Python alumni Eric Idle because everyone in tech is a nerd.

Q I see on the Raspberry Pi that there are multiple IDLEs. One is just called IDLE and the other is called IDLE 3. What does this mean?

A The standard IDLE uses Python 2, while IDLE 3 uses Python 3. They're both two slightly different versions of Python, with 3 having a few more and different functions. Due to the popularity of Python 2, though, it's still prevalent in tutorials and projects and included on the Raspberry Pi.

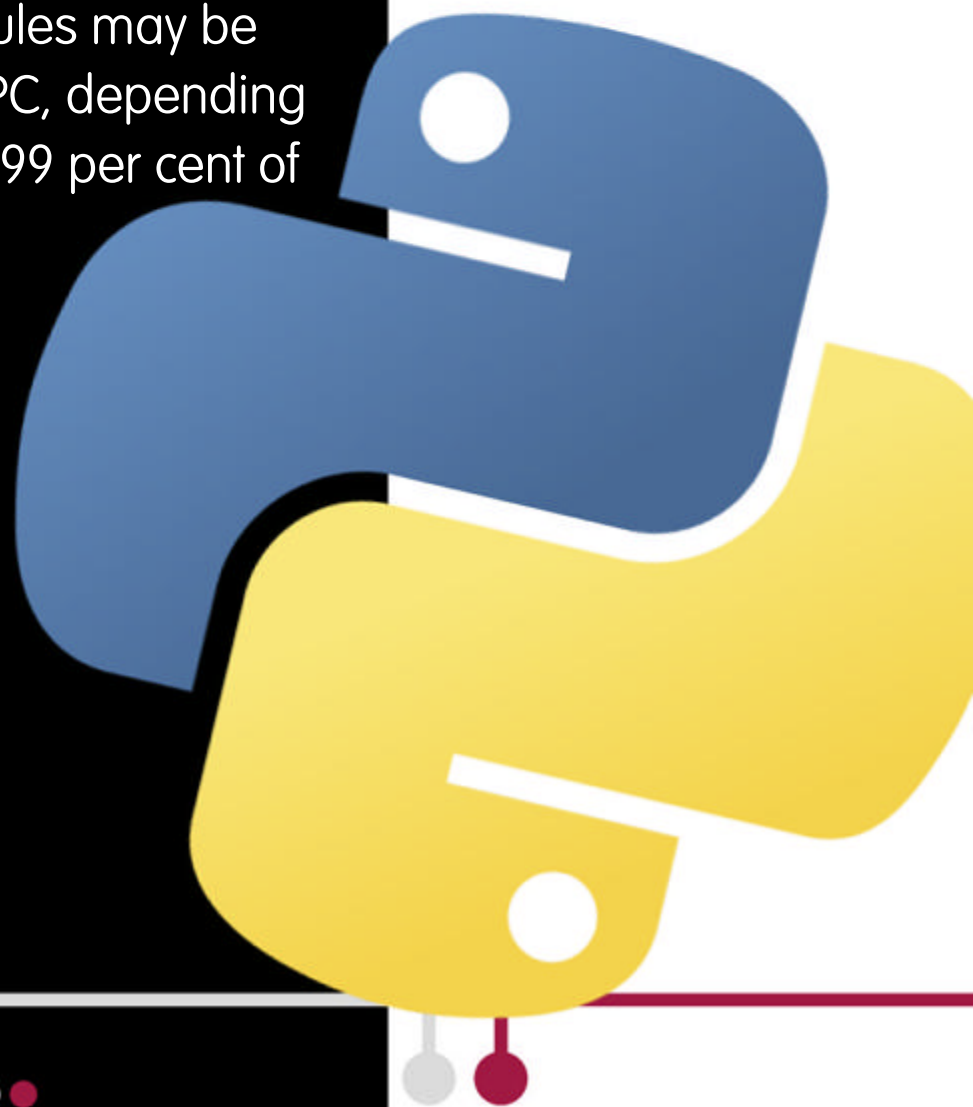
Q Which one should I be using?

A If you're just about to start learning, we'd recommend going for the newer Python 3. That said, going from 2 to 3 is not a huge deal, so when the big switchover finally occurs you'll be in a good position to learn the changes as you go if you do start with Python 2.

Q Will the code I create on the Raspberry Pi work elsewhere?

A The code is not platform-specific, so as long as you have the same modules and files on the PC you want to transfer it to it will work just fine. Some modules may be slightly different between different types of PC, depending on what's compatible on the Pi and PC, but 99 per cent of the time you shouldn't have a problem.

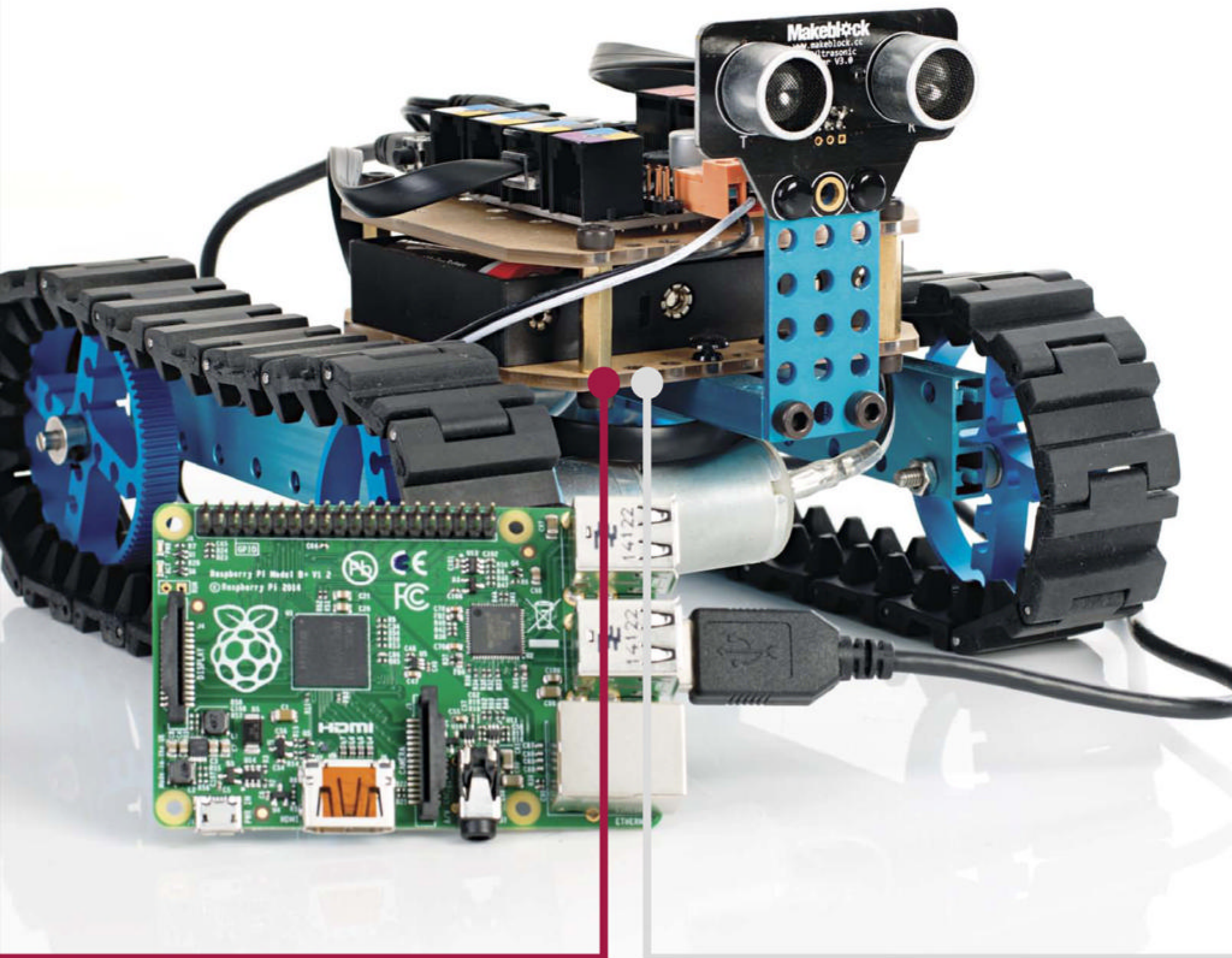
“The code is not platform-specific, so as long as you have the same modules and files then it will work just fine”





Program Arduino on the Raspberry Pi

You can interface a Raspberry Pi with Arduino components – now learn how to program them from the Pi and control a robot





Learning to code is one of the best things about owning a Raspberry Pi for a lot of people. Python and C are easy enough to start with on the Pi and you can get great results in a short time. When it comes to physical computing and making, though, not much beats using the Arduino IDE to control the open source controllers, servos and sensors associated with the system. Once set up, we can also use the Pi to program robots like the Makeblock, a great robotics starter kit, using either the built-in commands or your own code.



**Makeblock robot
starter kit**

[www.makeblock.cc/
starter-robot-kit-blue-ir-
version/](http://www.makeblock.cc/starter-robot-kit-blue-ir-version/)

01 Update your Pi

Always make sure your Raspberry Pi is up to date. This tutorial is based around Raspbian so set up an SD card if you don't have one, and then do an update with **sudo apt-get update && sudo apt-get upgrade**. Make sure the firmware is up to date as well with a `sudo rpi-update`.

02 Install Arduino IDE

Next you'll need to install Arduino to the Raspberry Pi – luckily it's in the repos for Raspbian so all you'll need to is open a terminal and type:

```
$ sudo apt-get install arduino
```

03 Install Arduino libraries

To get everything working on your Pi you'll also need to get the relevant libraries as well. These will allow you to connect to various Arduino boards, especially the Makeblock kit. Do this with:

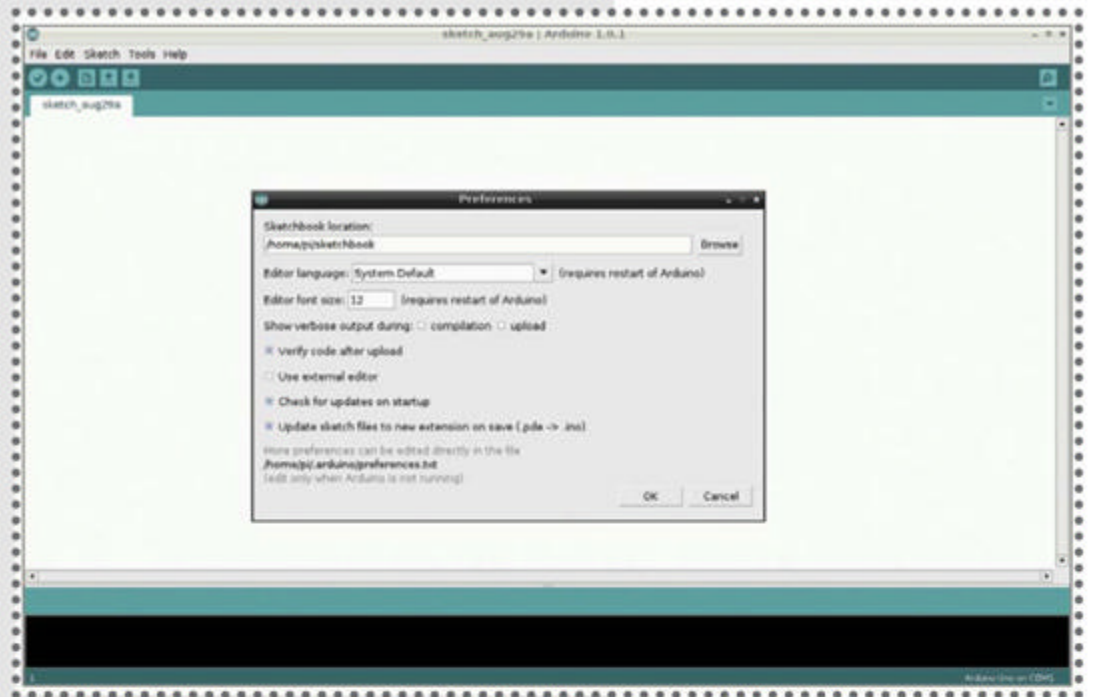
```
$ sudo apt-get install avr-libc libftdi1  
avrdude openjdk-6-jre librx-tx-java
```

“Not much beats using the Arduino IDE to control the open source controllers, servos and sensors associated with the system”



04 Using Arduino

Now everything is ready you can open up the Arduino IDE. It can be found in the new Electronics section of the programs menu. It will create a new sketchbook folder in the home menu where all your projects will live.



05 Makeblock starter kit libraries

To program the Makeblock, it's best to grab the example code so you can get an idea of how it works. The zip for these can be downloaded from the Makeblock GitHub page, which hosts all the official libraries: <http://bit.ly/1J7yvNH>. Unzip it to a separate folder before continuing.

06 Install Makeblock examples

Turn off the Arduino IDE, if you haven't already, and find the 'sketchbook' folder in the home directory. Create a new folder inside called libraries and copy and paste the 'makeblock' folder from the unzipped files into this new directory to effectively install it to Arduino.

07 Prepare your robot

Once you've built your Makeblock robot, unplug the battery and plug the control board into the Raspberry Pi via a micro USB cable. Restart the Arduino IDE and go to Tools, Board and select Arduino Leonardo. Also check to make sure the right Serial Port is selected.

“Once you've built your Makeblock robot, unplug the battery and plug the control board into the Raspberry Pi via a micro USB cable”



08 Program your robot

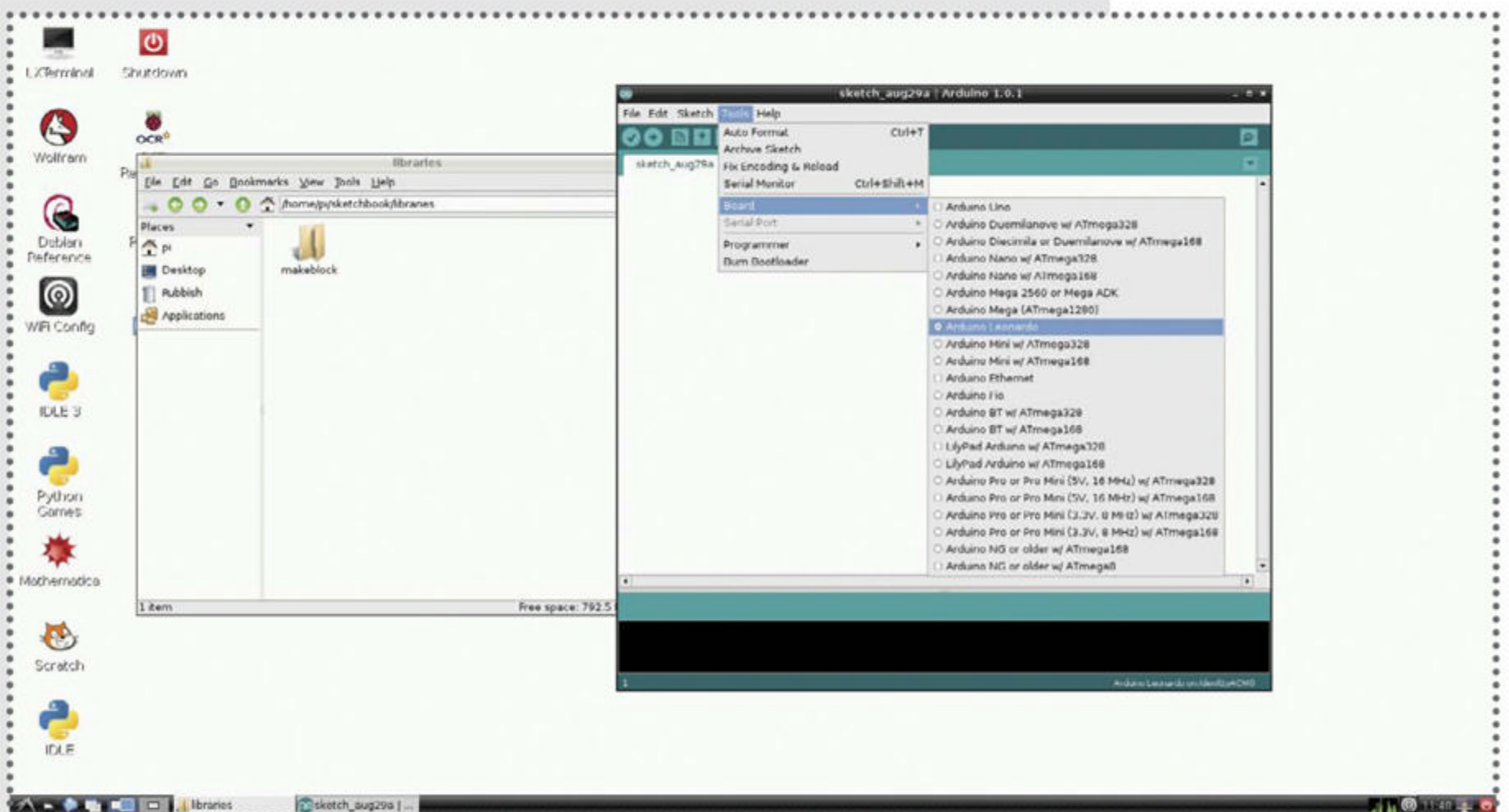
Now you can load an example code to make sure everything is working. Click on File, Examples and find the 'makeblock' menu. Go to the Starter Kit option and select IR Control example. Upload it to the Makeblock, unplug it from the Pi and then give it a go with the remote control.

09 More programming

Now you can modify the example codes to change the functionality of the Makeblock. This will allow you to understand how the robot, and other Arduino devices, are programmed so that you can start writing your own custom code for it.

“Now you can modify the example codes to change the functionality of the Makeblock”

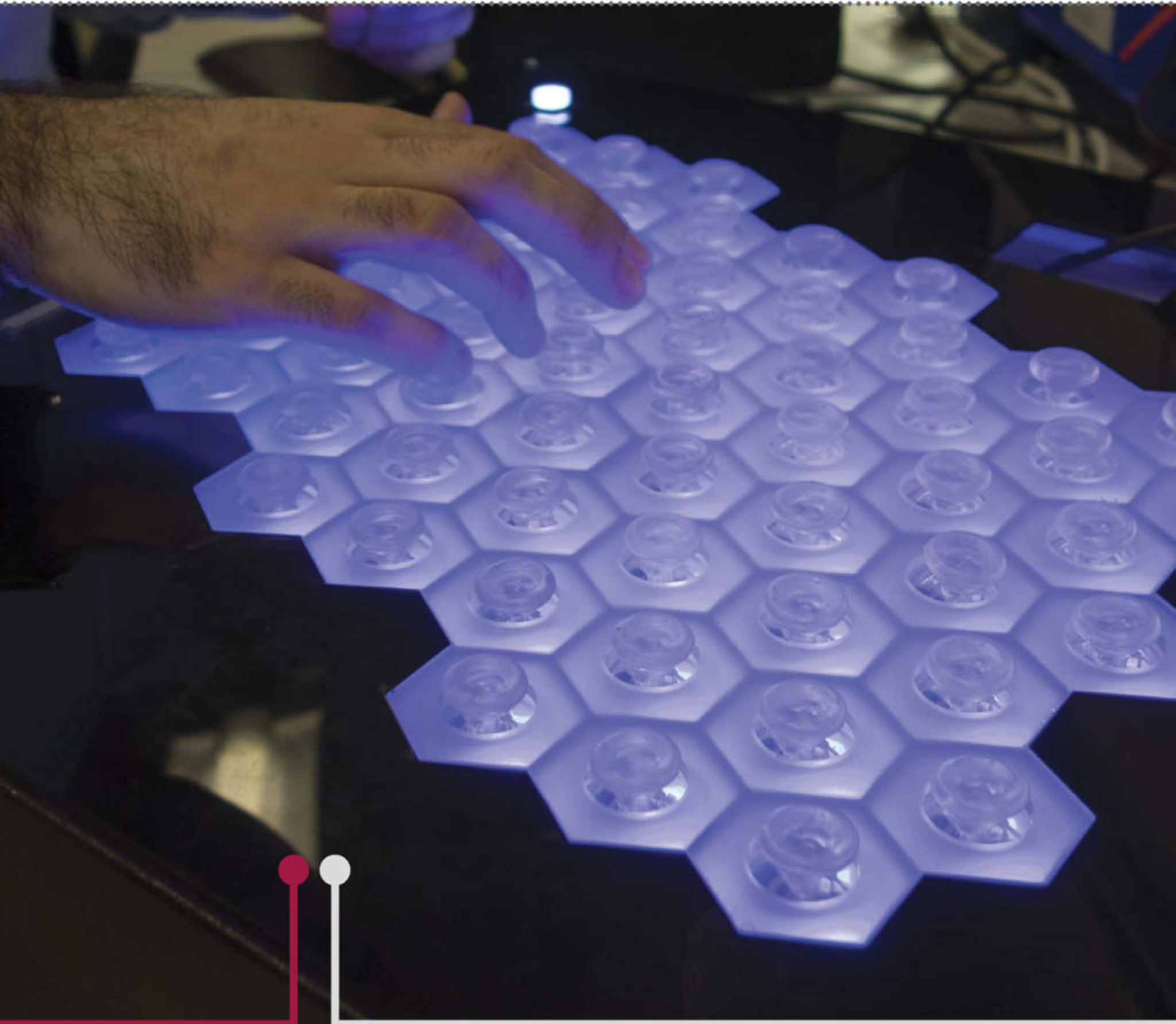
Below The Arduino IDE has some great presets for all the different kinds of Arduino board that are available





Joytone

David Sharples and David Glanzman reinvent the keyboard...
by replacing the keys with a hex grid of joysticks





What inspired the Joytone?

David Sharples So the Joytone is an expressive isomorphic musical instrument. The project came out of my frustration trying to learn to play musical instruments. I was trying to teach myself piano and I was wondering why there are white keys and black keys, you know, it makes things so much harder. I learned a little bit of musical theory and came to realise that a major scale is just a well-defined musical structure; the gaps between the notes in a major scale are the same regardless of which major scale it actually is. But, because there's white keys and black keys on a piano, you can play a C major scale just by running up all the white notes. If you want to play a C# major scale you have to throw in a whole bunch of black keys. It's hard to remember and you have to build up this huge muscle memory. So one of the big goals with this project is to build a musical instrument that doesn't have that bias based on the keys – so it's isomorphic; that's what that means.

And you're using analogue thumbsticks?

David Glanzman They're Xbox joysticks...

Sharples That's the second big goal of the project. When I was doing research about this, I noticed there were some instruments that had these hexagonally isomorphic keyboards – a grid of hexagons – but the issue was that they were just buttons, they didn't have the same sort of richness and depth as an actual musical instrument. When you're playing a guitar there are a million ways to affect the sound – how hard you're pushing on the string, how hard you pluck it, where on the fret you're holding it, if you bend the string or not – and you can get all these rich sonic qualities. So we wanted to make it isomorphic and we wanted to make it expressive. We used these thumbsticks because you get two channels



David Sharples

is an interaction designer and graduated from the University of Pennsylvania's Digital Media Design programme



David Glanzman

is a sophomore in Computer Engineering at the University of Pennsylvania. David has worked on microprocessor design, audio electronics, medical devices and more



of analogue control in this really familiar little interface. One axis changes the waveform of the synthesised sound from a triangle wave (has a pure, bell-like quality) to a reverse sawtooth wave (has a buzzy, bright sound, like a trumpet). There are two oscillators creating each note and if you push the thumbstick to the left, those oscillators are exactly in tune, making a very soft sound. If you push it all the way to the right then they're offset by a couple of hertz, which makes a wide, rich sound. Then the amount that you rotate the joystick, regardless of direction, gives the volume. So you have like two and half dimensions of control there, which adds some depth to the sound.

What is the role of the Raspberry Pi?

Sharples There's kind of a two-brain system going on – we have the Raspberry Pi and then we have the Cypress PSoC 4. The Cypress PSoC 4 does all the heavy lifting with the data reading.

Glanzman It does all the measurements for the joysticks. It's got ADCs in it that convert analogue to digital, and then basically looks at each axis for each joystick in sequence, records it, and then waits for the Raspberry Pi to ask it for data values for each of the joysticks.

Sharples There's 57 thumbsticks and each one has two analogue channels, so that's 114 analogue channels total. So what we did was we had eight 16-channel multiplexers hooked up to the PSoC and then the PSoC sends a signal to all of them that says 'give me channel one'. Then it reads the eight channels, and then it says 'give me channel two' and it reads the eight channel twos. After it does that for all 16 channels it then has this full bank of current data. The Raspberry Pi periodically says 'give me all your most recent data', so the PSoC forwards the data to the Raspberry Pi, which then does a little bit

If you like

Interested in learning more about isomorphic instruments?

Check out David Sharples' senior design blog:

<http://bit.ly/1dNQdli>

Further reading

Here's the final demo video of the Joytone, comprising David Glanzman's virtuoso Bach performance:

<http://bit.ly/1vfXnlw>



of processing in Python and then sends commands to PureData, which is our synthesiser program.

What's the Arduino component doing?

Sharpley Each thumbstick also has an RGB LED in its little hexagonal cell, and our intention was to use these to show which notes are in key or out of key. We also wanted to guide the user through a scale – or even a song, showing the next note that they're supposed to play – but we ran into some technical difficulties. The ideal setup for this is that you daisy-chain all of these lights in a big line and then you hook them up to an Arduino Micro, which is actually really good at controlling these specific lights from Adafruit, and then it can just push all of this data down the line and you should just be able to put whatever colours you want on each light individually. But we had some problems with the signal and could only get about four lights to behave.

Is it easy to learn to play the Joytone?

Sharpley The barrier to entry is much lower. It's funny, when we first got it working, David was playing with it, wearing headphones, and he sort of stopped talking and was pushing a few of the joysticks, like 'Wait, wait...', and then he just played back Bach. So the key to learning it is just learning a little, tiny bit about the structures in music theory. There's a shape you can hold your fingers in that will play a major chord; once you learn that shape, that's it – that's how all of the Joytone's major chords are played.

Glanzman When it comes to learning the Joytone, you have to attack musical instruction differently than you would with another instrument. When you learn something like the piano, you learn that this is D major,

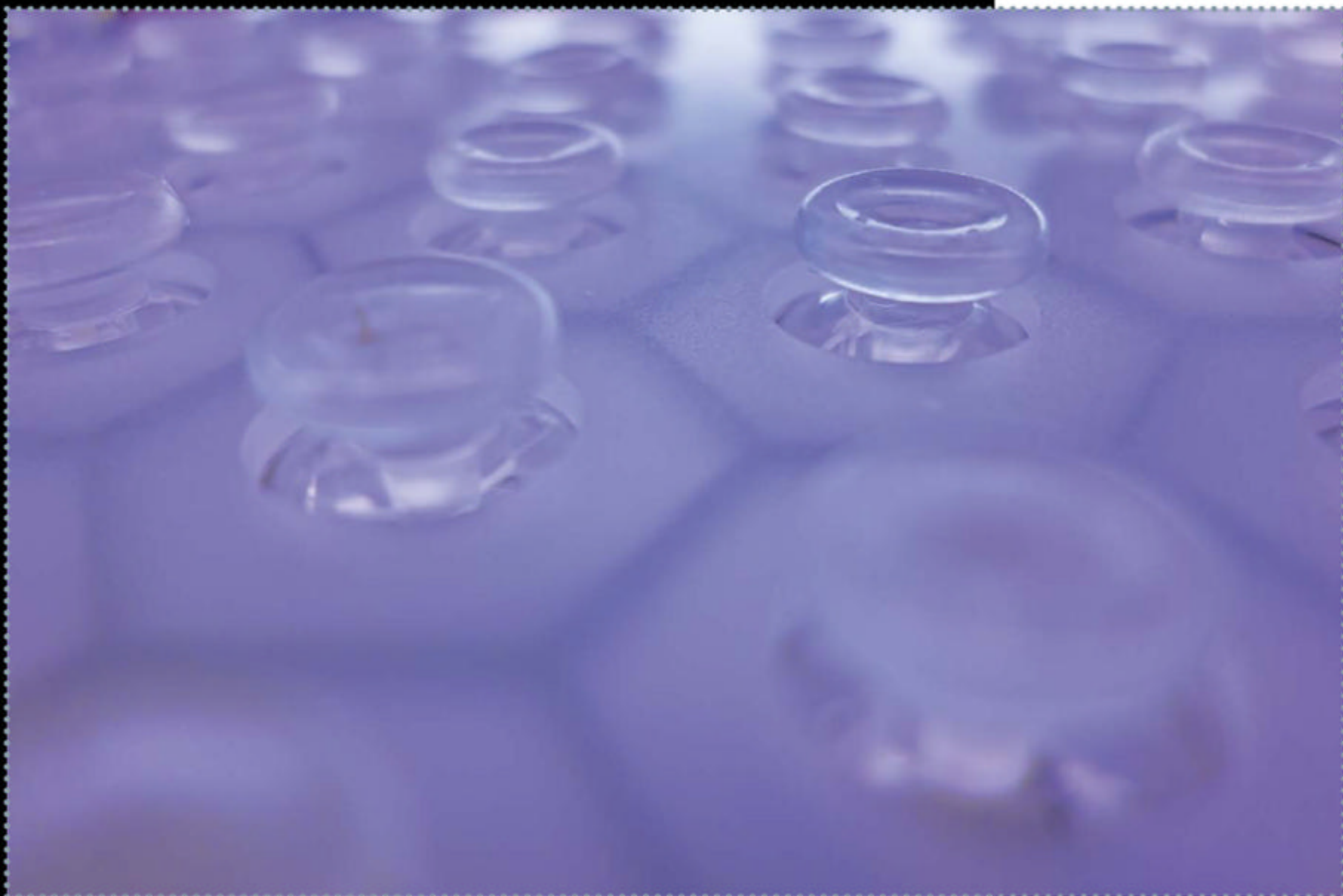


this is F# minor – you learn different things based on the note, the actual class, right?

But with the Joytone, the pitch class is totally irrelevant because we hear everything in relevant terms, and you play everything in relative terms. So to learn the instrument, you don't even have to discuss pitch classes – you just talk about relative distances. So major thirds or minor thirds, fifths, fourths – it's distances between notes instead of the actual note values.

I think if you phrase musical instruction in those terms, in terms that we experience music in rather than the terms we normally go through to create music, it becomes a much more natural interface with the Joytone because it's built on that type of instruction, making it simple to learn.

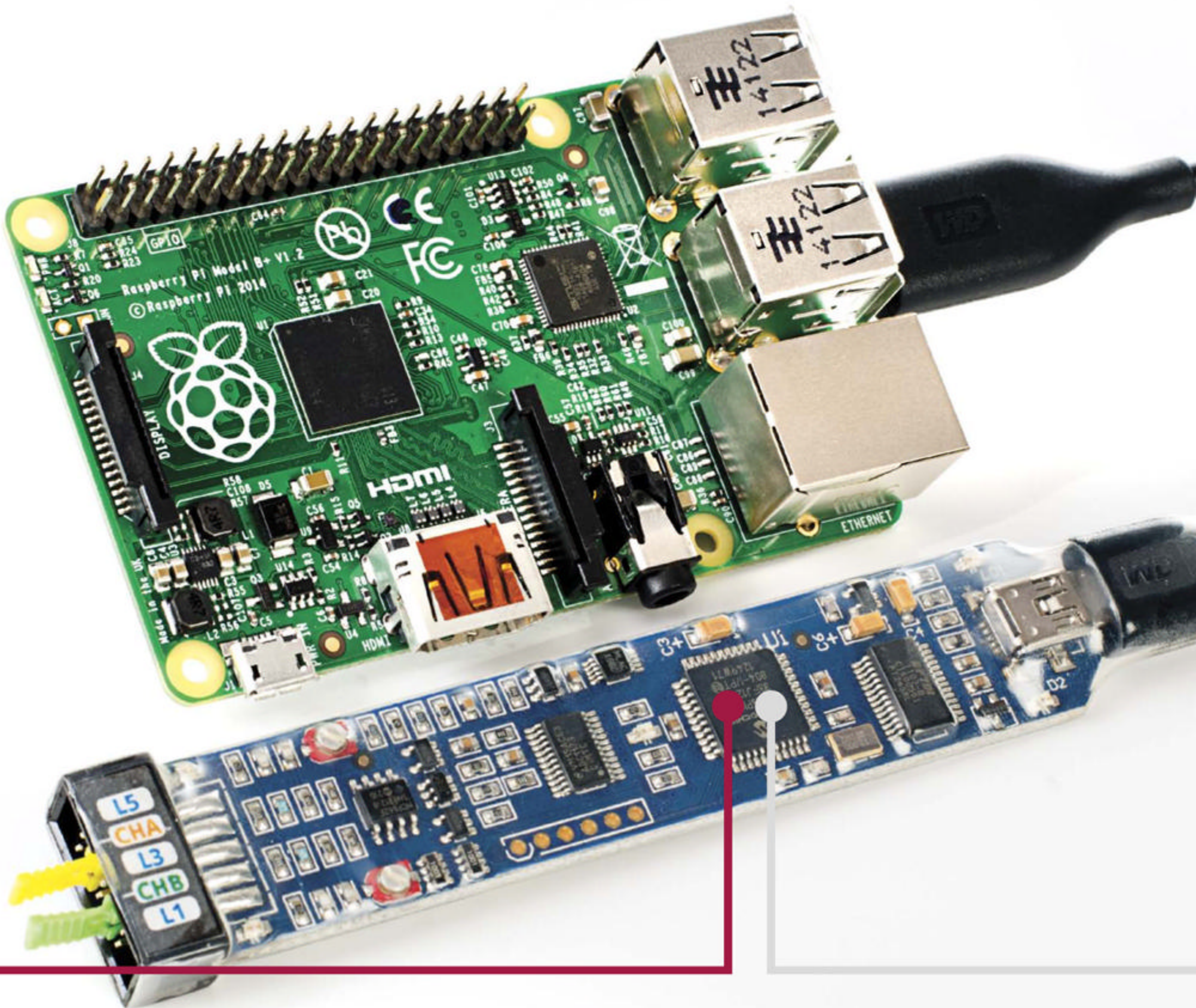
Below Getting to know the musical distances between adjacent thumbsticks is the key to learning to play the Joytone






Transform your Pi into a micro oscilloscope

Prepare to turn your Raspberry Pi into a fully functional micro oscilloscope, logic analyser and waveform generator with the BitScope Micro



The Raspberry Pi logo, which is a red circle containing several white circles of varying sizes arranged in a pattern resembling a raspberry.

The Raspberry Pi has been used in a plethora of applications in hardware, software and some quite unbelievable combinations of the two. From record-breaking space flights to automated bartending devices and much more, there are thousands of Raspberry Pi projects that, over the last three years, have shown what a capable little Linux box this is.

The BitScope Micro is certainly no exception and when you couple it with your Raspberry Pi you have a very powerful, pocket-sized oscilloscope that also features a whole host of other functionalities, such as a waveform and clock generator as well as a spectrum and logic analyser. Best of all though, the whole setup (including the Raspberry Pi itself) comes in at well under £150. Requiring no external power source, the BitScope Micro is also water resistant and so is perfect for either home or lab use. It is fully configurable and user programmable in Python, C++ and more, and can even continuously stream data to disk.

“Requiring no external power source, the BitScope Micro is also water resistant and so is perfect for either home or lab use”

Below To find out more about the BitScope Micro and order one now, go to: <http://bit.ly/1QG6wbX>

01 Grab your BitScope

If you have not already done so, you need to go and order your shiny new BitScope Micro (directly from BitScope or from one of their worldwide retailers). If you are serious about electronics then you need a good oscilloscope, so it is truly worth every penny! Once it arrives, you should be greeted with the neatly packaged box pictured to the right.



02 Open up the box

Once you have received your BitScope Micro and opened up the box for the first time you should find all of the pictured items inside (if you get any extras, then it is obviously your lucky day). The main contents are the BitScope Micro itself (with mini USB cable preattached) and a set of ten test clip grabbers. There is also a variety of documentation containing a large amount of product info and guidance.

03 Update your Raspberry Pi

As with almost every project you undertake on the Raspberry Pi, it pays dividends to make sure your operating system is updated to the latest stable version, as this can save a lot of hassle further down the line. To do this, open an LXTerminal session and then type:

```
sudo apt-get update  
sudo apt-get upgrade -y
```

Then wait patiently for the upgrade process to complete.

04 Locate the BitScope Software

Now your Raspberry Pi is all up to date you need to acquire the BitScope DSO (digital storage oscilloscope)

“The main contents are the BitScope Micro itself (with mini USB cable preattached) and a set of ten test clip grabbers”

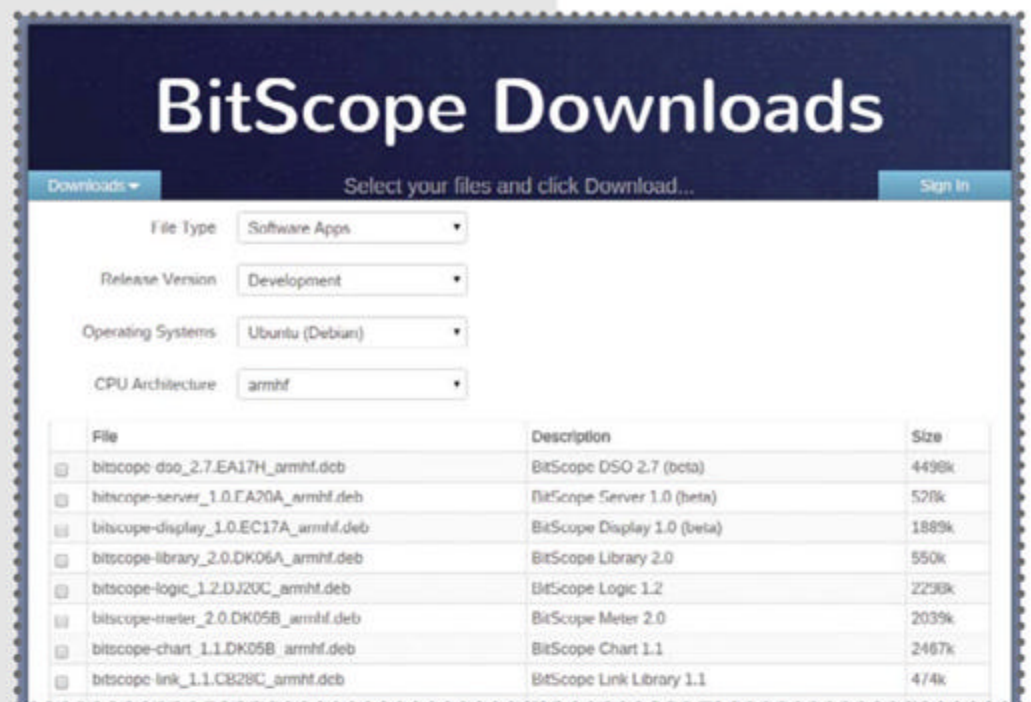


Left The BitScope software is available in the DEB and RPM formats for Debian and Red Hat-based distributions

software. This is not yet available as a Raspbian package, but it is very easy to install anyway using the downloadable DEB file. Visit www.bitscope.com/pi and click on the Download link at the top.

05 Download the software

The previous step should have brought you to the BitScope Downloads page. From here you need to download the top item in the list, BitScope DSO 2.7 (beta), and save it to the /home/pi directory on your Raspberry Pi so you know where to find it later. On some browsers the file will automatically download to the /home/pi/Downloads directory.



Above DSO is the only software under Production but there's a lot more available in the Development section of the page

06 Install the software

Now we have downloaded the package, the easiest way to install the software is to open an LXTerminal session and then run the following code...

```
sudo dpkg -i bitscope-dso_2.7.EA17H_armhf.deb
```

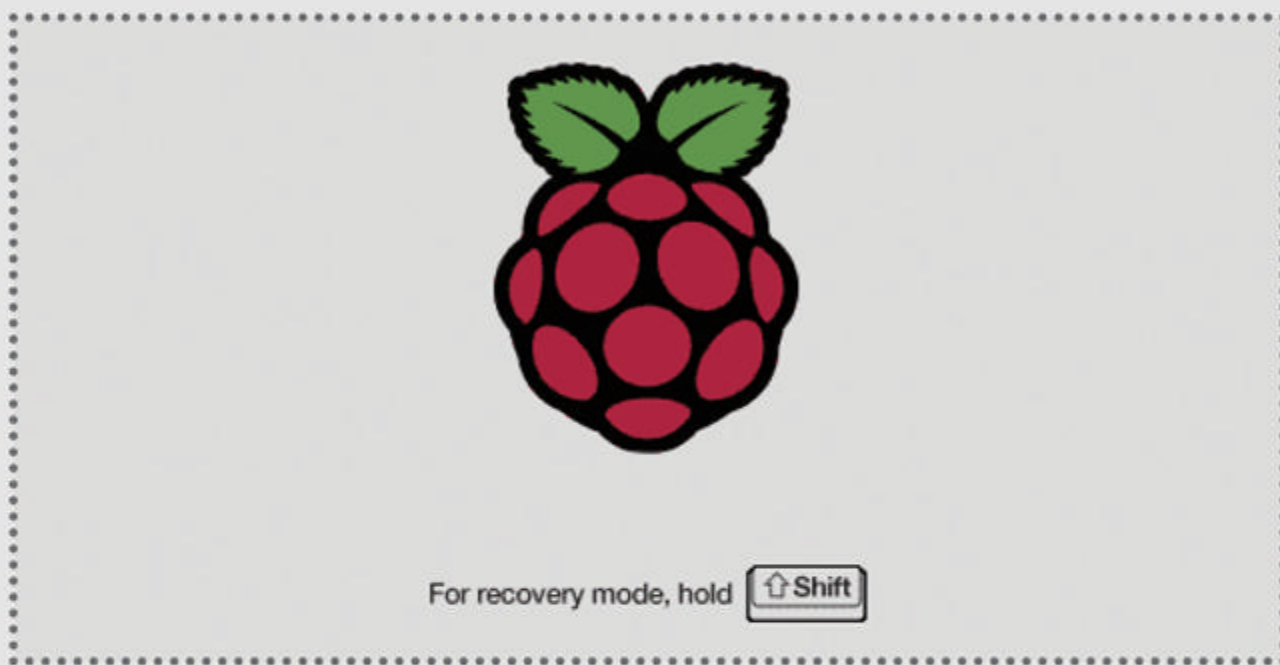
...or the equivalent version for newer software.

You should see lines of output as the installation progresses. The BitScope DSO software then appears in the main menu under Other.

07 Overclock your Raspberry Pi (optional)

The BitScope software will run fine on a Raspberry Pi with default settings, however it can be a bit slow to respond. Open an LXTerminal session and type sudo

raspi-config. In the menu, select option 7 Overclock. Click OK on the following screen and on the next one select Turbo. Click OK and then you should see some code run. Once this completes press OK and then you are brought back to the main raspi-config window. Select Finish at the bottom right, and then select Yes to reboot your Raspberry Pi.



Left Overclocking is a tricky business – you may need to enter recovery mode if your settings don't work first time

08 Overclocking – part two

Overclocking can sometimes cause instability on your Raspberry Pi or an inability to boot at all. If this happens, you can press and hold the Shift key on your keyboard once you reach the above splash screen to boot into recovery mode. You can then re-do step 7 at a lower overclock setting and repeat until you find the highest stable setting.

09 Plug in the BitScope

Now that the software has been successfully installed on your Raspberry Pi, we can get started with the BitScope. If you are using a Model A or B Raspberry Pi without a USB hub then I would recommend turning the Raspberry Pi off before attaching the BitScope or it may crash. The B+ should be fine with plug and play.

10 Load the BitScope DSO

Now you can navigate to the BitScope DSO software in the menu. This should load a splash screen with two options – POWER and SETUP. Click POWER and then OK on the pop-up information box. After a minute or less, the BitScope DSO main screen will load and you should see some lights on the BitScope start to flash.

11 Familiarise yourself with the software

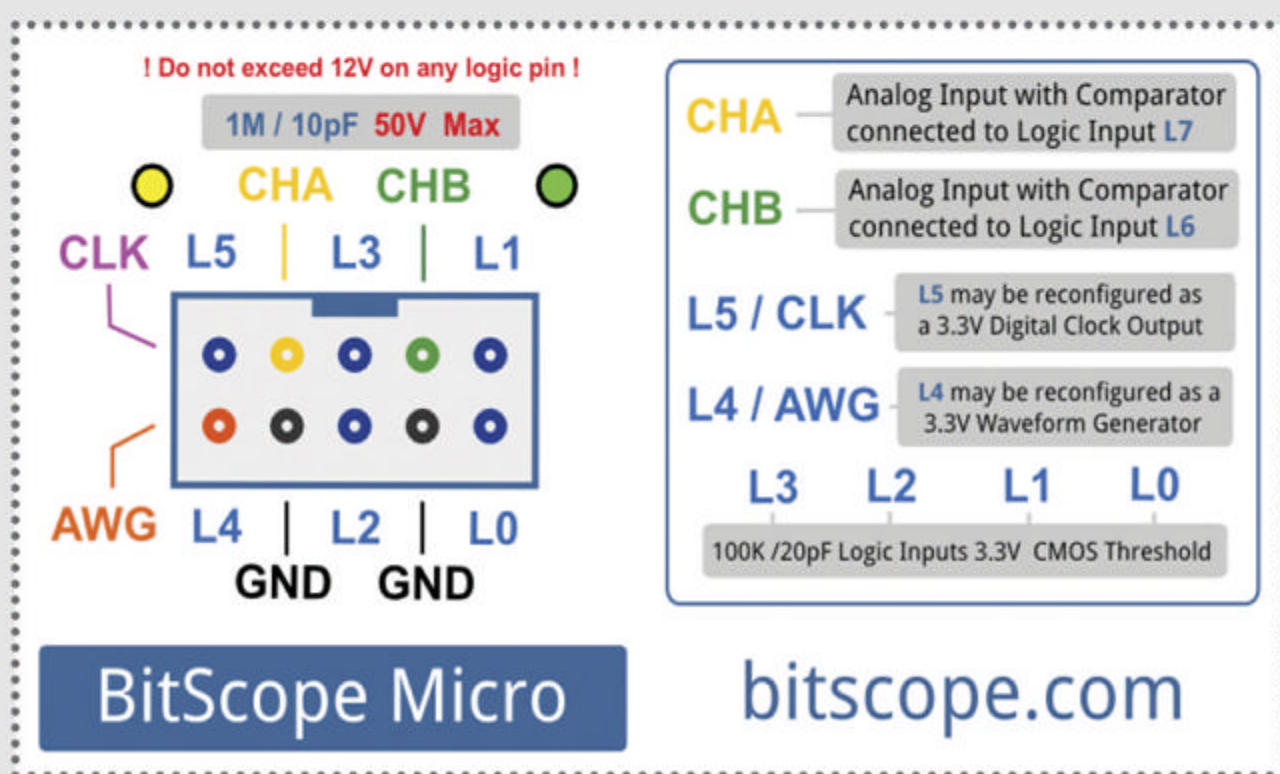
The screen layout of the BitScope DSO software is fairly intuitive, and is similar to other physical or virtual oscilloscopes. The largest part is the main display window. To the top-left is the trigger window, which changes to wave control if selected. Under the main window you have the analog input channels and various trim adjustments.

“To the top-left is the trigger window, which changes to wave control if selected.

Under the main window you have the analog input channels and various trim adjustments”

12 Familiarise yourself with pinout

The image below shows the BitScope Micro pinout diagram. There are a total of ten pins with two of them being ground pins (GND). The remaining eight pins can



Left Note the warning: do not exceed 12V on any of the logic pins!

be configured as logic pins and four of them also have different functions – L4 is also a waveform generator (AWG), L5 is a clock generator, (CLK) and L7 and L6 relate to CHA and CHB respectively.

13 Perform a sample experiment

The easiest way to test whether or not your BitScope is working correctly is to connect one of the test clip grabbers to the analog input CHA on the BitScope. Connect the other end to physical pin two on your Raspberry Pi and adjust the scale of the y axis to 2V/div. You should then see an output in the main window of around five volts.

14 Use different waveforms

The BitScope can generate its own waveforms. Connect a female-to-female jumper cable between CHA and L4 (AWG). On the right-hand side of the DSO interface, select Wave and then a wave should appear in the main screen. Change the x axis to 100us/Div and the y axis to 500mV/Div. Right-click on the yellow OFFSET button and select MEDIAN. The wave should now fill the main window as in the above screenshot. You can adjust various parameters of the waveform in the wave control panel top-left and can also change to step (square) or ramp (saw-tooth) waves instead of tone (sinusoidal) waves.

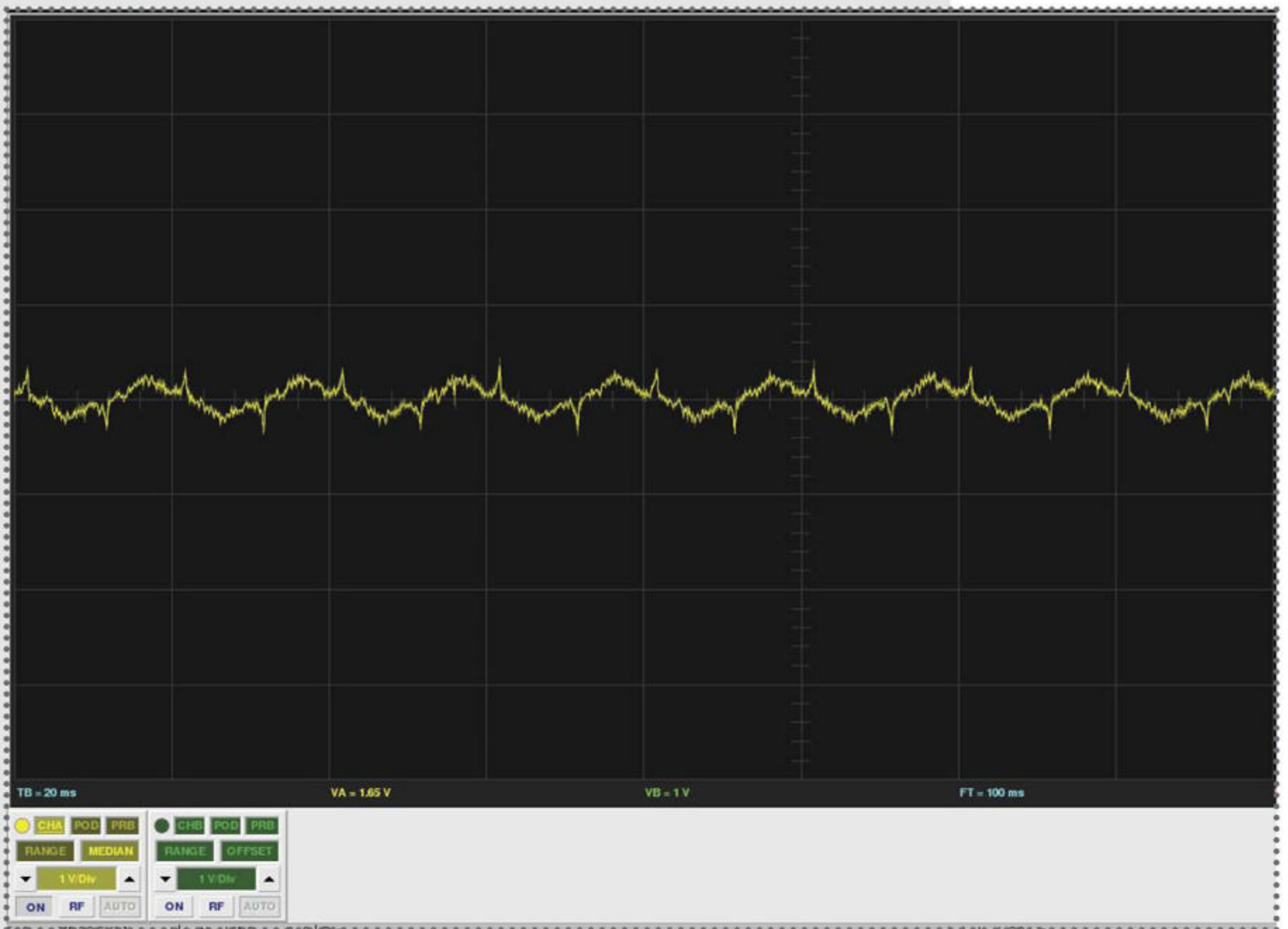
15 Experimenting with your body

Another interesting (but fairly simple) thing to try is measuring electrical signals from your body. Set the vertical axis to 1V/Div and horizontal to 20ms/Div. Then plug in one of the probes to CHA, pull back the grabber end and touch it with your finger. You should then see a

Hardware upgrades

One of the best things about the BitScope Micro is that it runs on exactly the same software as the more capable hardware in the range. This means if at some point in the future you feel the BitScope Micro is not enough for your needs, you can quickly and easily upgrade to better hardware with no hassle, and no need to learn any new software!



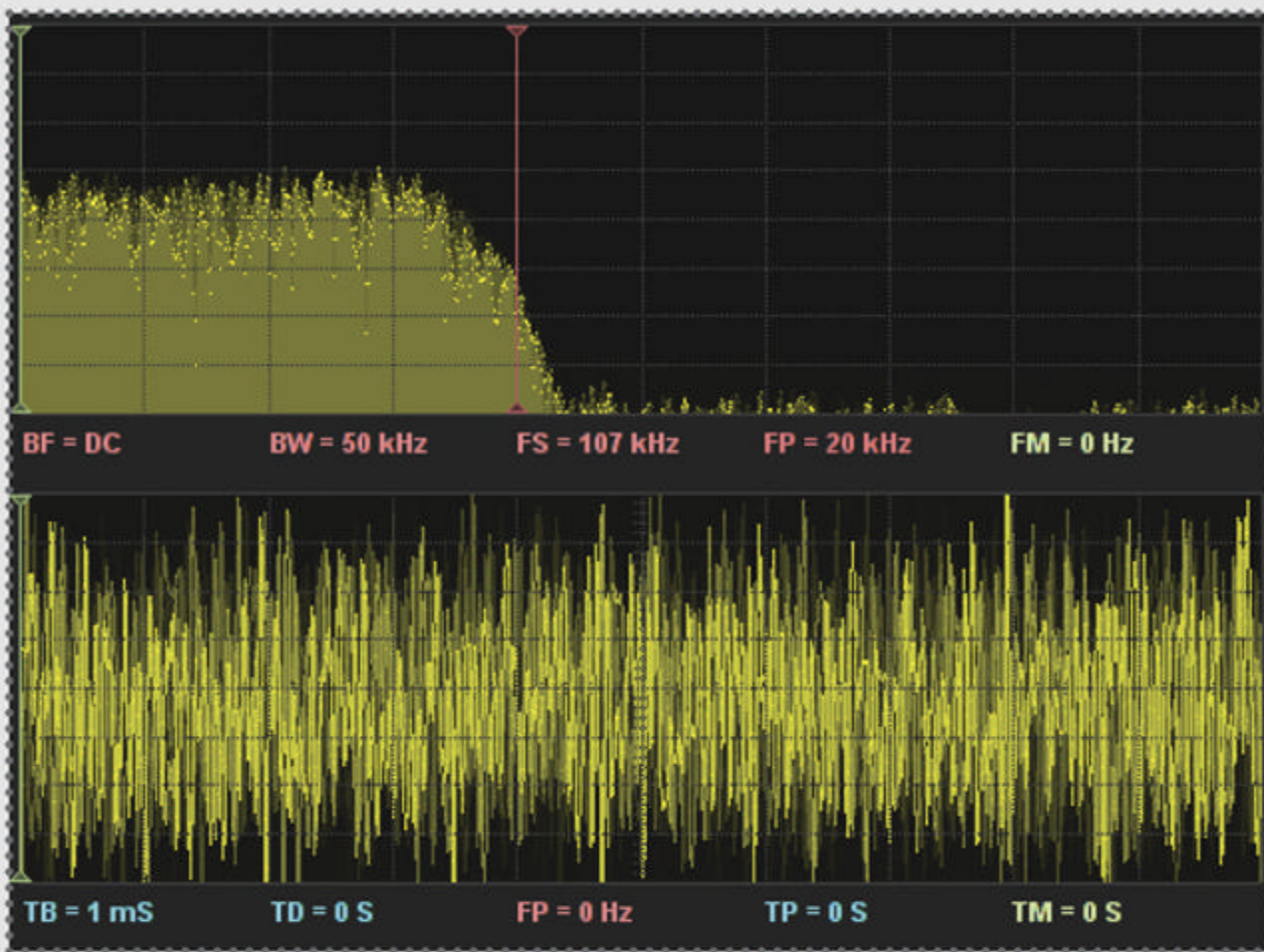


sine wave on the screen. Bizarrely, this wave is actually radio waves emitted by your mains electrical wiring which are then being picked up by your body (which is acting as an antenna). This is why the wavelength of the signal you see is approximately 50 to 60 Hz.

Above The BitScope is incredibly sensitive –try playing around with the on-screen options as you go

16 Programming your BitScope

The BitScope DSO and other available software (BitScope Logic and BitScope Chart) are very powerful applications for a lot of experimentation. However, in some more complex monitoring environments they may not offer enough flexibility. The good news is that you can program the BitScope in C/C++, Python or Pascal using their programming API.

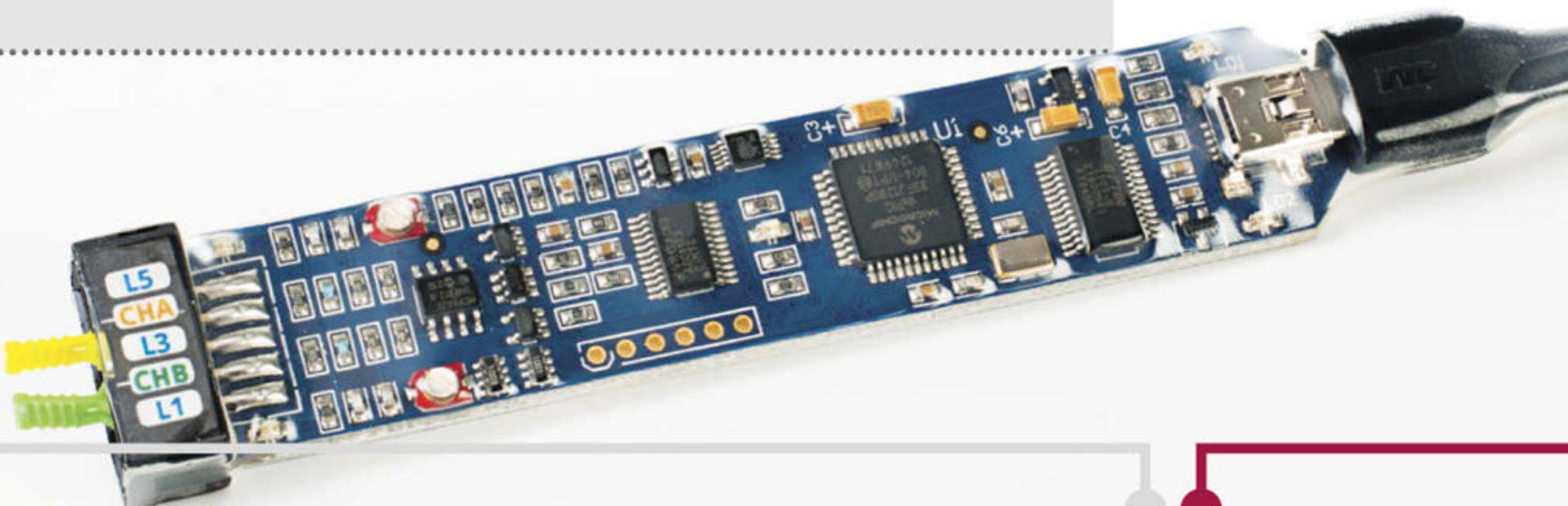


Left Here you can see white noise being passed through a 20 kHz audio amplifier

17 Further experiments to try

This tutorial has shown only a small fraction of what the BitScope Micro is capable of. As seen in the above image it can also be used as a spectrum analyser along with a whole host of other functionality. Perhaps for your next experiment you could measure the resistance of your body by comparing the voltage drop across your body with that of a known resistor. Or you could also try probing your I2C or SPI connections in order to see how they work. If you ever run out of ideas then why not visit the BitScope website and start experimenting further!

Below The BitScope Micro is also incredibly portable, so it's great for throwing into your kit bag





Talking Pi

Join the conversation at...



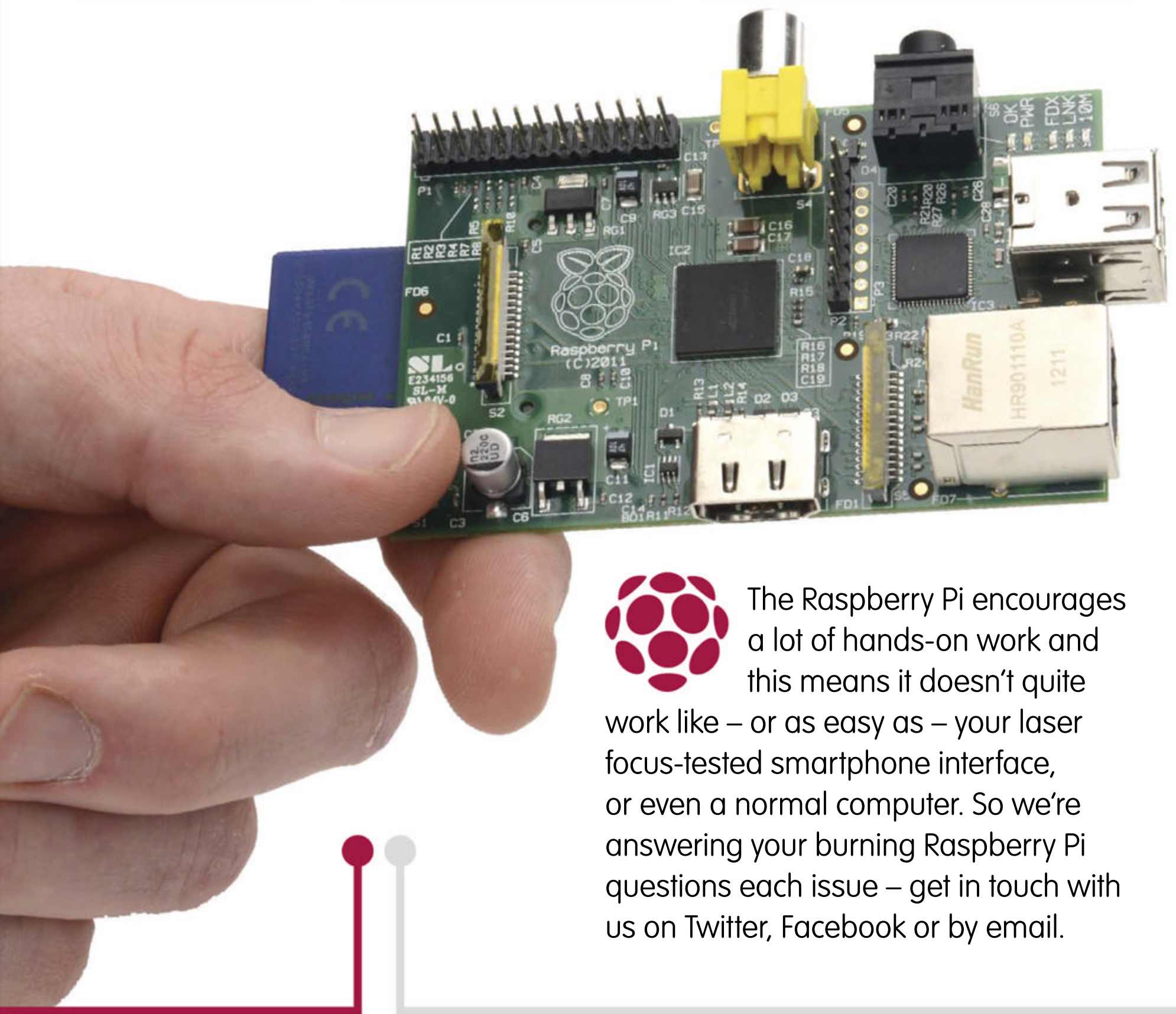
@linuxusermag



Linux User & Developer



RasPi@imagine-publishing.co.uk



The Raspberry Pi encourages a lot of hands-on work and this means it doesn't quite work like – or as easy as – your laser focus-tested smartphone interface, or even a normal computer. So we're answering your burning Raspberry Pi questions each issue – get in touch with us on Twitter, Facebook or by email.

Should I get the official Raspberry Pi case?

Laura via email

The new official case is an excellent product, but if you've already got a case for your Raspberry Pi then there's no reason to switch it out for this new one. That being said, if you find some of the other cases a little tricky to use for GPIO and camera placement, you can snap the top off this case pretty easily to access them, and it is of course a lovely-looking case. It all really depends on what you want out of your Raspberry Pi case.



Keep up with the latest Raspberry Pi news by following @LinuxUserMag on Twitter. Search for the hashtag #RasPiMag

JUST A SCORE
WHAT'S YOUR JUST A SCORE?

Have you heard of Just A Score? It's a new, completely free app that gives you all the latest review scores. You can score anything in the world, like and share scores, follow scorers for your favourite topics and much more. And it's really good fun!

Can I only use one SD card per Raspberry Pi?

Tod via Facebook

You can use as many different SD cards with your Raspberry Pi as you like – although only one can be used at a time. This can be very useful, as you can have multiple SD cards for different projects and uses. A general desktop card, for example, one to control a robot and then a third for when you need a media PC! Just make sure you keep the software up to date on all of them, including firmware updates.



Is there going to be a Raspberry Pi 2 Model A?

Filippo via Twitter

While the Raspberry Pi Foundation haven't announced that there will be one, they also said last year that there wouldn't be a Raspberry Pi 2 for many years to come, so we're being optimistic about the Foundation's propensity for nice surprises. If it is announced, there'll be very little forewarning – the Model A stuff is usually announced at the end of the summer time though, going by previous years, so don't give up on the Model 2A until the end of the year. It may even be February's next big announcement.



Can I link two Raspberry Pis together using the GPIO ports to increase their power?

Vanessa via Facebook

No, not really; that's not how the GPIO ports work. You can definitely use two different Raspberry Pis in a project that involves some physical computing and have them split up certain aspects of the work, but the best way to try and pool their power together is over the network using supercomputer-style code to share resources as they compute tasks together – check out the first few pages of this issue for our guide to setting up a Pi cluster.



JUST A
SCORE
WHAT'S YOUR JUST A SCORE?

You can score absolutely anything on Just A Score. We love to keep an eye on free/libre software to see what you think is worth downloading...

10 LinuxUserMag scored 10 for Keybase

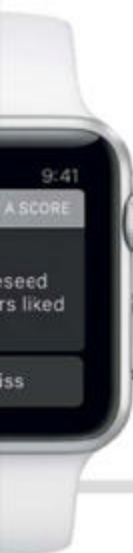
9 LinuxUserMag scored 9 for Cinnamon Desktop

8 LinuxUserMag scored 8 for Tomahawk

4 LinuxUserMag scored 4 for Anaconda installer

3 LinuxUserMag scored 3 for FOSS That Hasn't Been Maintained In Years

SCORE ANYTHING
JUST A SCORE

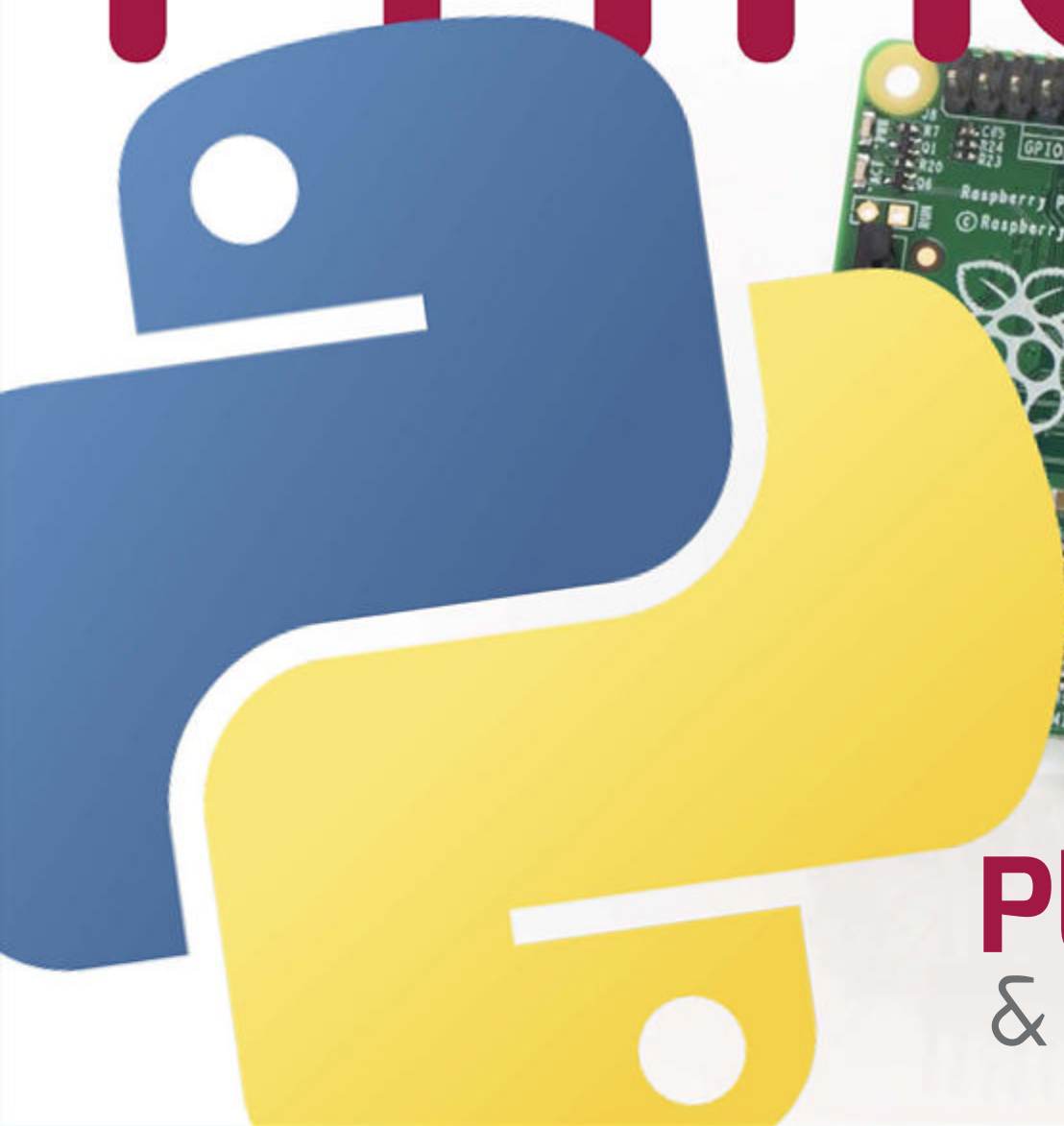




Next issue

 Get inspired  Expert advice  Easy-to-follow guides

WHY YOU NEED PYTHON 3



Plus Photo frames
& wireless speakers

Get this issue's source code at:
www.linuxuser.co.uk/raspicode